

UNIVERSITÉ DU QUÉBEC

**MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES**

**PAR
FANO ANDRIAMIHARISOA**

**ANALYSE ET COMPRÉHENSION DES SUITES DE TEST :
UNE ÉTUDE EMPIRIQUE SUR PLUSIEURS
SYSTÈMES ORIENTÉS-OBJET**

JUILLET 2014

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

**ANALYSIS AND UNDERSTANDING OF TEST SUITES:
AN EMPIRICAL STUDY ON SEVERAL
OBJECT ORIENTED SYSTEMS**

Fano Andriamiharisoa

ABSTRACT

Testing is a very important activity in the development process of software systems. Costs generally associated with the testing process and verification and validation in general are exorbitant. Any way to estimate (or predict), ideally early in the development process, the difficulty of creating the test suites, guide the testing process in order to increase its effectiveness and identify the critical parts of a software system has a significant impact on the overall costs of development and maintenance, not to mention the quality of the product.

Software metrics can be used to estimate software testability, one of the most important attributes of software quality. The testing coverage is a result of the application of tests. By analyzing the test suites of several systems, we try to determine how we can optimize the testing effort while covering a large extent of a system during the testing process. We propose through this work a federative approach to prioritize tests by using models to predict the testing effort and testing coverage.

Our experimental approach consists in an observation of object-oriented metrics values over testing coverage, an observation of the testing effort and testing coverage, and the application of a statistical prediction method, multivariate logistic regression, in order to optimize the testing effort and maximize the testing coverage.

REMERCIEMENTS

Ma plus profonde reconnaissance va avant tout à mon Créateur, Jéhovah Dieu, dont la bonté et la bienveillance m'ont aidé à mener à terme ce mémoire.

Je remercie également mes professeurs Monsieur Mourad Badri et Madame Linda Badri pour leur encadrement de proximité et leur générosité tant intellectuelle que financière.

Je tiens à remercier tout particulièrement ma douce épouse Miora. Ta compréhension et tes encouragements ont été inspirants tout au long de ces années d'études.

À mes chers parents, merci pour votre soutien sans faille depuis le début de cette aventure.

Enfin, je souhaite remercier toutes les autres personnes qui ont participé de différentes façons à la réussite de mon mémoire, en l'occurrence mes frères et sœurs, mes proches amis et mes collègues étudiants.

Cette étude a été rendue possible grâce à la contribution financière du CRSNG (Conseil de Recherches en Sciences Naturelles et en Génie du Canada) et à la fondation de l'UQTR.

ANALYSE ET COMPRÉHENSION DES SUITES DE TEST : UNE ÉTUDE EMPIRIQUE SUR PLUSIEURS SYSTÈMES ORIENTÉS-OBJET

Fano Andriamiharisoa

SOMMAIRE

Les tests représentent une activité très importante du processus de développement des systèmes logiciels. Les coûts généralement associés au processus de test et à la vérification et validation en général sont exorbitants. Tout moyen permettant d'estimer (voire prédire), idéalement tôt dans le processus de développement, la difficulté de créer les jeux de tests, les orienter dans le but d'augmenter leur efficacité (ou leur portée) et de déterminer les parties critiques d'un système logiciel, a une forte incidence sur les coûts globaux de développement et de maintenance, sans compter la qualité du produit.

Les métriques logicielles peuvent être utilisées pour estimer la testabilité, un des attributs les plus importants de la qualité des logiciels. La couverture de test quant à elle est une résultante de l'application des tests. En analysant les suites de tests de plusieurs systèmes, nous cherchons essentiellement à déterminer comment peut-on optimiser l'effort de test tout en couvrant dans une large mesure un système lors du processus de test. Nous proposons à travers ce mémoire une approche fédérative pour prioriser les tests en se servant de modèles de prédiction de l'effort de test et de la couverture des tests.

Notre démarche expérimentale a consisté en une observation des valeurs des métriques orientées-objet par rapport à la couverture des tests, une observation portant sur l'effort de test et la couverture des tests, et l'application d'une méthode statistique de prédiction, la régression logistique multivariée, pour l'optimisation de l'effort de test et la maximisation de la couverture des tests.

TABLE DES MATIÈRES

CHAPITRE 1. Introduction.....	9
1.1. Problématique.....	10
1.2. Approche	11
1.3. Organisation	12
CHAPITRE 2. État de l'art : bref aperçu	14
2.1. Métriques orientées-objet	14
2.1.1 Métriques de cohésion	14
2.1.2 Métriques de couplage	15
2.1.3 Métriques d'héritage	17
2.1.4 Métriques de taille.....	18
2.1.5 Métriques de complexité.....	18
2.2. Métriques de test	20
2.3. Méthodes d'expérimentation	22
2.3.1 Corrélations.....	22
2.3.2 Régression logistique	23
2.3.3 K-Means.....	24
2.4. Outils d'expérimentation	24
2.4.1 Borland Together	25
2.4.2 XLSTAT	25
2.4.3 Weka	25
CHAPITRE 3. Catégorisation et Taux de couverture	26
3.1. Taux de couverture.....	26
3.1.1 Définition	26
3.1.2 Limitation.....	27
3.1.3 Re-calcul	29
3.2. Catégorisation des classes orientées-objet	31
CHAPITRE 4. Observation des métriques orientées-objet versus la couverture de tests.....	33
4.1. Objectif.....	33
4.2. Hypothèse.....	33
4.3. Systèmes logiciels	35
4.3.1 JFREECHART.....	35
4.3.2 JODA	36
4.3.3 IO	36
4.3.4 POI	37
4.4. Expérimentation	37
4.4.1 JFREECHART.....	38
4.4.2 JODA	55
4.4.3 IO	72
4.4.4 POI	88

CHAPITRE 5.	Observation de l'effort de test versus la couverture de tests	105
5.1.	Objectif.....	105
5.2.	Hypothèse.....	105
5.3.	Expérimentation	107
5.3.1	JFREECHART.....	107
5.3.2	JODA	112
5.3.3	IO	116
5.3.4	POI	120
CHAPITRE 6.	Vers la priorisation des tests	126
6.1.	Objectif.....	126
6.2.	Hypothèse.....	127
6.3.	Démarche expérimentale.....	127
6.4.	Expérimentation	131
6.4.1	Régression logistique.....	131
6.4.2	Interprétation des modèles	134
6.4.3	JFREECHART.....	137
6.4.4	JODA	141
6.4.5	IO	144
6.4.6	POI	147
CHAPITRE 7.	Conclusion.....	152
Bibliographie.....		155

LISTE DES TABLEAUX

Tableau 1 - Échantillon de valeurs du taux de couverture	27
Tableau 2 - Données du tableau 1 après re-calcul du taux de couverture	29
Tableau 3 - Répartition des 3 différentes catégories	32
Tableau 4 - Statistiques descriptives des données des classes de CAT-2	38
Tableau 5 - Statistiques descriptives des données des classes de CAT-1	39
Tableau 6 - Statistiques descriptives des données des classes de CAT-0	39
Tableau 7 - Corrélations de Spearman des métriques OO des classes de CAT-2.....	42
Tableau 8 - Corrélations de Spearman des métriques OO des classes de CAT-1.....	43
Tableau 9 - Corrélations de Pearson des métriques OO des classes de CAT-2	43
Tableau 10 - Corrélations de Pearson des métriques OO des classes de CAT-1	44
Tableau 11 - Synthèse de la classification k-means.....	52
Tableau 12 - Correspondance des catégories et les clusters.....	53
Tableau 13 - Statistiques descriptives des données des classes de CAT-2	56
Tableau 14 - Statistiques descriptives des données des classes de CAT-1	56
Tableau 15 - Statistiques descriptives des données des classes de CAT-0	57
Tableau 16 - Corrélations de Spearman des métriques OO des classes de CAT-2....	60
Tableau 17 - Corrélations de Spearman des métriques OO des classes de CAT-1	60
Tableau 18 - Corrélations de Pearson des métriques OO des classes de CAT-2	61
Tableau 19 - Corrélations de Pearson des métriques OO des classes de CAT-1	61
Tableau 20 - Synthèse de la classification k-means.....	69
Tableau 21 - Correspondance des catégories et les clusters.....	70
Tableau 22 - Statistiques descriptives des données des classes de CAT-2	73
Tableau 23 - Statistiques descriptives des données des classes de CAT-1	73
Tableau 24 - Statistiques descriptives des données des classes de CAT-0	74
Tableau 25 - Corrélations de Spearman des métriques OO des classes de CAT-2....	76
Tableau 26 - Corrélations de Spearman des métriques OO des classes de CAT-1	77
Tableau 27 - Corrélations de Pearson des métriques OO des classes de CAT-2	77
Tableau 28 - Corrélations de Pearson des métriques OO des classes de CAT-1	78
Tableau 29 - Synthèse de la classification k-means.....	85
Tableau 30 - Correspondance des catégories et les clusters.....	86
Tableau 31 - Statistiques descriptives des données des classes de CAT-2	89
Tableau 32 - Statistiques descriptives des données des classes de CAT-1	89
Tableau 33 - Statistiques descriptives des données des classes de CAT-0	90
Tableau 34 - Corrélations de Spearman des métriques OO des classes de CAT-2....	93
Tableau 35 - Corrélations de Spearman des métriques OO des classes de CAT-1	93
Tableau 36 - Corrélations de Pearson des métriques OO des classes de CAT-2	94
Tableau 37 - Corrélations de Pearson des métriques OO des classes de CAT-1	94
Tableau 38 - Synthèse de la classification k-means.....	101
Tableau 39 - Correspondance des catégories et les clusters.....	102
Tableau 40 - Statistiques descriptives des données des métriques de tests.....	108
Tableau 41 - Corrélations de Spearman : métriques de tests / taux de couverture ..	108
Tableau 42 - Corrélations de Pearson : métriques de tests / taux de couverture.....	108
Tableau 43 - Statistiques descriptives des données des métriques de tests.....	113
Tableau 44 - Corrélations de Spearman : métriques de tests / taux de couverture ..	113

Tableau 45 - Corrélations de Pearson : métriques de tests / taux de couverture.....	114
Tableau 46 - Statistiques descriptives des données des métriques de tests.....	116
Tableau 47 - Corrélations de Spearman : métriques de tests / taux de couverture ..	117
Tableau 48 - Corrélations de Pearson : métriques de tests / taux de couverture.....	117
Tableau 49 - Statistiques descriptives des données des métriques de tests.....	120
Tableau 50 - Corrélations de Spearman : métriques de tests / taux de couverture ..	121
Tableau 51 - Corrélations de Pearson : métriques de tests / taux de couverture.....	121
Tableau 52 - Équations de modèle obtenues pour les 4 systèmes.....	135
Tableau 53 - Échantillon de données de prédiction de JFREECHART.....	139
Tableau 54 - Statistiques : effort de test faible - couverture des tests élevée.....	140
Tableau 55 - Statistiques : reste des classes.....	140
Tableau 56 - Échantillon de données de prédiction de JODA	142
Tableau 57 - Statistiques : effort de test faible - couverture des tests élevée.....	143
Tableau 58 - Statistiques : reste des classes.....	143
Tableau 59 - Échantillon de données de prédiction de IO	145
Tableau 60 - Statistiques : effort de test faible - couverture des tests élevée.....	146
Tableau 61 - Statistiques : reste des classes.....	146
Tableau 62 - Échantillon de données de prédiction de POI	148
Tableau 63 - Statistiques : effort de test faible - couverture des tests élevée.....	149
Tableau 64 - Statistiques : reste des classes.....	149

LISTE DES FIGURES

Figure 1 - Variation du taux de couverture vs la métrique COC (CAT-2)	46
Figure 2 - Variation du taux de couverture vs la métrique COC (CAT-1)	47
Figure 3 - Variation du taux de couverture vs la métrique FO (CAT-2)	47
Figure 4 - Variation du taux de couverture vs la métrique FO (CAT-1)	47
Figure 5 - Variation du taux de couverture vs la métrique DOIH (CAT-2).....	48
Figure 6 - Variation du taux de couverture vs la métrique DOIH (CAT-1).....	48
Figure 7 - Variation du taux de couverture vs la métrique LCOM (CAT-2).....	48
Figure 8 - Variation du taux de couverture vs la métrique LCOM (CAT-1).....	49
Figure 9 - Variation du taux de couverture vs la métrique LOC (CAT-2).....	49
Figure 10 - Variation du taux de couverture vs la métrique LOC (CAT-1).....	49
Figure 11 - Variation du taux de couverture vs la métrique WMPC (CAT-2)	50
Figure 12 - Variation du taux de couverture vs la métrique WMPC (CAT-1)	50
Figure 13 - Variation du taux de couverture vs la métrique COC (CAT-2)	64
Figure 14 - Variation du taux de couverture vs la métrique COC (CAT-1)	64
Figure 15 - Variation du taux de couverture vs la métrique FO (CAT-2)	65
Figure 16 - Variation du taux de couverture vs la métrique FO (CAT-1)	65
Figure 17 - Variation du taux de couverture vs la métrique DOIH (CAT-2).....	65
Figure 18 - Variation du taux de couverture vs la métrique DOIH (CAT-1).....	66
Figure 19 - Variation du taux de couverture vs la métrique LOC (CAT-2).....	66
Figure 20 - Variation du taux de couverture vs la métrique LOC (CAT-1).....	66
Figure 21 - Variation du taux de couverture vs la métrique WMPC (CAT-2)	67
Figure 22 - Variation du taux de couverture vs la métrique WMPC (CAT-1)	67
Figure 23 - Variation du taux de couverture vs la métrique COC (CAT-2)	80
Figure 24 - Variation du taux de couverture vs la métrique COC (CAT-1)	80
Figure 25 - Variation du taux de couverture vs la métrique FO (CAT-2)	81
Figure 26 - Variation du taux de couverture vs la métrique FO (CAT-1)	81
Figure 27 - Variation du taux de couverture vs la métrique DOIH (CAT-2).....	81
Figure 28 - Variation du taux de couverture vs la métrique DOIH (CAT-1).....	82
Figure 29 - Variation du taux de couverture vs la métrique LOC (CAT-2).....	82
Figure 30 - Variation du taux de couverture vs la métrique LOC (CAT-1).....	82
Figure 31 - Variation du taux de couverture vs la métrique WMPC (CAT-2)	83
Figure 32 - Variation du taux de couverture vs la métrique WMPC (CAT-1)	83
Figure 33 - Variation du taux de couverture vs la métrique COC (CAT-2)	96
Figure 34 - Variation du taux de couverture vs la métrique COC (CAT-1)	96
Figure 35 - Variation du taux de couverture vs la métrique FO (CAT-2)	97
Figure 36 - Variation du taux de couverture vs la métrique FO (CAT-1)	97
Figure 37 - Variation du taux de couverture vs la métrique DOIH (CAT-2).....	97
Figure 38 - Variation du taux de couverture vs la métrique DOIH (CAT-1).....	98
Figure 39 - Variation du taux de couverture vs la métrique LOC (CAT-2).....	98
Figure 41 - Variation du taux de couverture vs la métrique WMPC (CAT-2)	99
Figure 42 - Variation du taux de couverture vs la métrique WMPC (CAT-1)	99
Figure 43 - Variation du taux de couverture vs la métrique de test Tass.....	110
Figure 44 - Variation du taux de couverture vs la métrique de test Tdata.....	110
Figure 45 - Variation du taux de couverture vs la métrique de test Tinvok.....	111

Figure 46 - Variation du taux de couverture vs la métrique de test Tloc	111
Figure 47 - Variation du taux de couverture vs la métrique de test Tass	115
Figure 48 - Variation du taux de couverture vs la métrique de test Tdata	115
Figure 49 - Variation du taux de couverture vs la métrique de test Tinvok.....	115
Figure 50 - Variation du taux de couverture vs la métrique de test Tloc	115
Figure 51 - Variation du taux de couverture vs la métrique de test Tass	118
Figure 52 - Variation du taux de couverture vs la métrique de test Tdata	118
Figure 53 - Variation du taux de couverture vs la métrique de test Tinvok.....	119
Figure 54 - Variation du taux de couverture vs la métrique de test Tloc	119
Figure 55 - Variation du taux de couverture vs la métrique de test Tass	122
Figure 56 - Variation du taux de couverture vs la métrique de test Tdata	122
Figure 57 - Variation du taux de couverture vs la métrique de test Tinvok.....	123
Figure 58 - Variation du taux de couverture vs la métrique de test Tloc	123

CHAPITRE 1. INTRODUCTION

Les tests représentent une activité importante et très coûteuse du processus de développement (et du cycle de vie d'une manière générale) des systèmes logiciels. Ils constituent, cependant, une tâche essentielle dans l'assurance de leur qualité. Les tests s'effectuent en plusieurs étapes successives (tests unitaires, tests d'intégration, tests système, tec.). L'effort et le coût reliés au processus de test croissent de manière significative avec la complexité des systèmes et la sévérité des exigences en matière de qualité. Les coûts généralement associés à ce processus de vérification et validation sont importants. Ils peuvent atteindre (voire dépasser), dans plusieurs cas, la moitié du coût total de développement. Il est donc clair que le coût du processus de test doit pouvoir être contrôlé et maîtrisé.

Tout moyen permettant d'estimer (voire prédire), idéalement tôt dans le processus de développement, la difficulté de créer les jeux de test, les orienter dans le but d'augmenter leur efficacité (ou leur portée) et de déterminer les parties critiques d'un système logiciel, a une forte incidence sur les coûts globaux de développement et de maintenance, sans compter la qualité. Ceci justifie grandement que les concepteurs de systèmes soient capables d'estimer a priori, dans un projet donné, les performances et les coûts relatifs au test. Ces différents éléments caractérisent la capacité d'un système à être testé et sont reliés à un attribut de qualité important appelé la testabilité.

La testabilité est, en effet, un des attributs les plus importants de la qualité des logiciels. La norme ISO la définit comme une des caractéristiques de la

maintenabilité. L'évaluation de la testabilité d'un système logiciel doit permettre de maîtriser le coût du test et d'apporter une aide dans la conduite du projet (estimation de l'effort de test, affectation des ressources, orientation du processus de test, détermination des composants sur lesquels un effort important est requis, etc.). Même s'il y a eu plusieurs publications sur le sujet (à différents niveaux), le concept reste difficile à cerner. Il est effectivement affecté par plusieurs facteurs.

Les métriques logicielles peuvent être utilisées pour estimer la testabilité d'un système (ou de ses composants). Des données quantitatives sur la testabilité d'un système peuvent, en effet, aider les concepteurs, testeurs et gestionnaires de projets à : planifier les activités de test, déterminer les parties critiques sur lesquelles ils devraient concentrer plus d'efforts pour assurer la qualité, et dans certains cas utiliser ces données pour piloter des activités d'inspection de code.

La couverture de test quant à elle est une résultante de l'application des tests et entre dans la recherche de la qualité des logiciels. L'atteinte d'une couverture maximale, va de pair avec la nécessité de développer des tests en conséquence et doit prendre en considération l'effort de test requis à cet effet.

1.1. Problématique

La recherche de la qualité est une quête impliquant, comme nous l'avons énoncé précédemment, un aspect économique qui doit être sérieusement pris en considération. Les coûts y afférents portent généralement sur le processus de test. Dans ce processus, l'effort de test doit être maîtrisé et contrôlé pour optimiser les coûts. Les métriques logicielles sont un moyen pour quantifier cet

effort de test et une manière de maîtriser et contrôler l'effort de test est de l'estimer assez tôt ou en d'autres termes, le prédire.

La couverture de test est une résultante de l'application des tests tel que le stipulent Mockus, A., Nagappan, N. et Dinh-Trong, D. T. [Mockus 09] : un cas de test d'une classe donnée peut affecter la couverture d'autres classes et d'autre part cet effet tend à se dissoudre par rapport au fait que le couplage soit faible au niveau des classes (ayant des cas de test).

Dans ce cas, comment optimiser l'effort de test tout en couvrant dans une large mesure un système lors du processus de test ?

1.2. Approche

Ce mémoire fait suite à deux sujets de recherches distincts mais liés. Tout d'abord Badri, M., Kout, A., Toure, F., [Kout 11] ont fait une analyse empirique d'un modèle de testabilité dont l'approche est basée sur les métriques orientées-objet. L'article en question propose de prédire la testabilité des programmes logiciels dès la phase de conception, ce qui a clairement une incidence sur l'optimisation des tests. Cela s'inscrit dans l'optimisation de l'effort de test recherchée dans le cadre de cette étude.

Par ailleurs, les travaux de Mockus, A., Nagappan, N. et Dinh-Trong, D. T. [Mockus 09] abordent l'éventualité que pour les classes faisant un nombre important d'appels, il est plus difficile d'atteindre un haut niveau de couverture, ce qui ouvre une piste de recherche sur les relations entre les métriques

orientées-objet (en l'occurrence celles qui touchent les appels) et la couverture des tests.

L'enjeu qui consiste à optimiser l'effort de test tout en cherchant une couverture importante des tests est de taille. Une démarche axée sur des observations minutieuses des caractéristiques de données et de l'application de méthodes de prédiction fiables et efficaces est requise; les données manipulées devant elles, être suffisamment hétérogènes pour valider la démarche empirique choisie.

1.3. Organisation

Le présent mémoire est divisé en cinq grandes parties. La première partie verra l'exposé d'un bref état de l'art dans lequel seront succinctement présentés les métriques orientées-objet, les métriques de test, les systèmes à l'étude, les méthodes et les outils d'expérimentation utilisés. Cette mise en contexte offrira un aperçu des différentes notions utilisées tout au long de notre étude et les cadres de recherche dans lesquels elles s'inscrivent.

La détermination du taux de couverture et la catégorisation des classes orientées-objet sont deux notions abordées tout au long du mémoire. De ce fait, la deuxième partie leur sera dédiée.

La troisième partie consistera en une observation des valeurs des métriques orientées-objet par rapport à la couverture des tests, dans le but de dégager à partir d'une analyse approfondie portant sur les données de métriques orientées-objet, des éléments caractéristiques frappants permettant de dresser

si possible un pattern (ou patron) établissant un lien entre les valeurs des métriques orientées-objet et la couverture des tests.

La quatrième partie s'apparente à la troisième partie, à la différence que les observations porteront sur l'effort de test (via les métriques de tests) et la couverture des tests, ce qui permettra de mieux saisir le lien entre l'effort de test et le taux de couverture.

La dernière partie abordera l'optimisation de l'effort de test et la maximisation de la couverture des tests, le tout pour arriver à prioriser les tests.

Une conclusion globale clôturera ce mémoire laissant une ouverture sur des questions à explorer dans le cadre de travaux à venir.

CHAPITRE 2. ÉTAT DE L'ART : BREF APERÇU

Dans ce chapitre, nous exposons un bref aperçu de l'état de l'art portant sur les métriques orientées-objet, les métriques de test, les méthodes et les outils utilisés dans le cadre des recherches relatives à la qualité logicielle, surtout dans le contexte des applications industrielles complexes qui se développent de nos jours. Un constat qui fait l'unanimité est que l'intérêt de l'évaluation de la qualité des programmes, avec leur complexité grandissante et les nombreux enjeux qui y sont rattachés, ne cesse de grandir.

2.1. Métriques orientées-objet

La littérature propose une grande variété de métriques dont cinq principaux groupes ayant des aspects liés à la qualité des systèmes orientés-objet, à savoir : la cohésion, le couplage, l'héritage, la taille et la complexité [Badri 09, Chidamber 94, Aggarwal 06]. On parle ici d'attributs internes de base qui ont une incidence sur les attributs externes de la qualité (tels que la testabilité).

2.1.1 Métriques de cohésion

- Lack of Cohesion in Methods (LCOM) mesure la dissimilarité des méthodes d'une classe en regardant les variables d'instance ou attributs utilisés par les méthodes [Chidamber 94]. Elle est définie comme suit : Soit P le nombre de paires de méthodes ne partageant aucun attribut commun et Q le nombre de méthodes qui partagent communément un attribut $LCOM = |P| - |Q|$, si $|P| > |Q|$. Si la différence est négative

LCOM est égale à 0. LCOM est une métrique que nous retiendrons dans le cadre de nos expérimentations.

- TCC (Tight Class Cohesion) : TCC donne le nombre de paires de méthodes qui partagent une variable d'instance d'une classe [Bieman 95]. Soit n , le nombre de méthodes d'une classe C donnée, $NP(C)$ le nombre de paires de méthodes est égale à $n(n-1)/2$, et $ND(C)$ le nombre de paires de méthodes accédant directement à la même variable d'instance : TCC est obtenu par $NP(C) / ND(C)$.

La cohésion est une mesure de la force des liens entre les méthodes des classes composant un module logiciel. Dans les systèmes orientés-objet, une forte cohésion est une propriété très recommandée [Larman 03] : en effet, il est largement reconnu que des composants fortement cohésifs tendent à avoir une maintenabilité et réutilisabilité significatives [Li 93, Bieman 95, Basili 96, Briand 97, Chae 00] ; la cohésion d'un composant informe sur la qualité de sa structure et le degré de cohésion d'un composant est élevé si ce dernier implémente une simple fonction logique.

2.1.2 Métriques de couplage

- Coupling between Objects (CBO) mesure le nombre d'autres classes auxquelles une classe est couplée [Chidamber 94]. Deux classes sont dites couplées si l'une d'elles utilise des méthodes ou des variables d'instance définies par l'autre classe. Plus le CBO est faible pour une classe, plus cette classe est facilement réutilisable. L'effort de test d'une

classe devrait être d'autant plus élevé que le CBO de cette classe est élevé [Kout 11].

- MIC (Method Invocation Coupling): MIC indique le nombre de classes vers lesquelles une classe donnée fait des appels.
- Fan Out (FO) qui est aussi une métrique de couplage se définit plus comme étant le nombre d'autres classes référencées par une classe : on parle plus communément d'« appels sortants » [Somerville 04].
- Fan In (FI, notée aussi COC) qui est aussi une métrique de couplage se définit comme étant le nombre classes qui référencent une classe donnée : ici, on parlerait d'« appels entrants » [Nasseri 10].

Ainsi, deux classes sont dites couplées si une méthode de l'une utilise une méthode ou un attribut de l'autre. Un couplage excessif entre classes se fait au détriment de la modularité et empêche la réutilisation [Briand 99]. Pour améliorer la modularité et favoriser l'encapsulation, le couplage entre les classes doit être réduit [Larman 03]. Par ailleurs, plus le couplage est élevé, plus le système est affecté par les changements (c'est ce qu'on appelle le « ripple-effect » traduit par « répercussions ») et plus la maintenance est difficile. Enfin, la mesure du couplage est un excellent indicateur de la testabilité [Kout 11].

2.1.3 Métriques d'héritage

- Depth of Inheritance Tree (DIT) est la profondeur de la classe dans l'arbre d'héritage. En cas d'héritage multiple, DIT vaudra la profondeur maximale depuis la classe racine jusqu'à la classe mesurée [Chidamber 94].
- Number of Children (NOC) est le nombre de descendants immédiats de la classe dans la hiérarchie des classes. Toutefois, la métrique NOC ne donne qu'une vue partielle, qui dans la majorité des cas sera une bonne approximation [Somerville 04].

L'héritage fournit une information conséquente sur l'arbre d'héritage d'un système. Il est un élément clé dans le paradigme orienté-objet. Ce mécanisme traduit la conception et la hiérarchie des classes et capture la relation entre une super classe et ses sous classes. Ainsi, plus une classe se situe profondément dans l'arbre d'héritage, plus le nombre de méthodes et de comportements, dont elle hérite, sera grand. Ceci rendra son comportement d'autant moins prévisible. Or, plus le nombre d'enfants est élevé plus la réutilisation effective est importante, car l'héritage est une forme de réutilisation, et donc, plus le risque d'une utilisation inadéquate du mécanisme de l'héritage est grand. Le nombre d'enfants donne une idée de l'influence potentielle d'une classe sur la conception. Si une classe a un nombre important d'enfants, elle pourra requérir d'avantage d'efforts de test. Enfin, l'héritage permet une meilleure réutilisabilité du code.

2.1.4 Métriques de taille

- Lines of Code (LOC) est une métrique logicielle utilisée pour mesurer la taille d'un logiciel en comptant le nombre de lignes dans le texte du code source du programme. Une classe large, en termes de lignes de code, est difficile à réutiliser, à comprendre et à maintenir, et présente un risque plus élevé de fautes [El Emam 01].
- Number of Attributes per Class (NOA) est une métrique de taille qui compte le nombre total d'attributs définis dans une classe [Henderson-Sellers 96]. Une classe possédant un grand nombre d'attributs pourrait indiquer la présence d'une cohésion fortuite, ce qui nécessite une autre décomposition afin de mieux gérer la complexité du modèle.
- Number of Operations of a Class (NOO) est une métrique de taille qui compte le nombre d'opérations définie dans une classe [Henderson-Sellers 96]. Si une classe possède un grand nombre d'opérations, celle-ci devient difficile à réutiliser. Ceci provoque souvent une perte de cohésion [Badri 09].

Les mesures de l'aspect « taille » sont intuitivement un indicateur de l'effort pour écrire le code [De Tran-Cao 05].

2.1.5 Métriques de complexité

- Response for a Class (RFC) RFC représente la cardinalité de l'ensemble de réponses d'une classe [Chidamber 94]. L'ensemble de réponses d'une

classe est l'ensemble des méthodes qui peuvent être directement appelées lors de l'exécution de n'importe quelle méthode de cette classe (réponse à un message). Si un grand nombre de méthodes peuvent être appelées en réponse à un message, le test et le débogage d'une classe deviennent plus compliqués, car ils requièrent un niveau de compréhension plus élevé de la part du testeur. La complexité d'une classe est d'autant plus élevée qu'elle peut appeler un grand nombre de méthodes et par la même, requérir un plus grand effort de test [Kout 11].

- Weighted Methods Per Class (WMC) est la somme pondérée des complexités des méthodes d'une classe [Chidamber 94]. Les classes qui ont un grand nombre de méthodes complexes sont, en général, plus spécifiques à l'application et donc moins réutilisables. Enfin, la complexité cyclomatique est un bon indicateur des risques d'erreurs, car une complexité élevée est synonyme de risque élevé de fautes dans la classe et de difficulté de compréhension de la classe [Basili 96, El Emam 01].

La complexité est un terme qui englobe de nombreuses propriétés d'un composant logiciel qui affectent toutes les interactions internes. En effet, elle décrit les interactions entre un certain nombre d'entités. Plus le nombre d'entités augmente, plus le nombre d'interactions entre elles augmente de façon exponentielle, pour arriver à un point où il serait impossible de connaître et de comprendre chacune d'elles. De même, un niveau élevé de la complexité dans un logiciel accroît le risque d'interférer avec les interactions et augmente donc les chances d'introduire des fautes lors des changements [De Tran-Cao 05].

Les métriques décrites précédemment sont toutes disponibles dans Together [Together]. Comme énoncé précédemment, cet outil nous a servi durant nos expérimentations pour calculer et analyser les valeurs des différentes métriques sélectionnées.

2.2. Métriques de test

L'écriture et le développement des tests JUnit résultent sur des classes de test correspondant chacune aux classes orientées-objet testées. Afin de capturer la testabilité des classes (du moins en termes d'effort lié à l'écriture des tests), la mesure de certaines caractéristiques des classes de test est nécessaire, et pour cela, nous utiliserons les métriques suivantes :

- TLoc : mesure le nombre de lignes de code d'une classe de test. Si une classe de test contient un nombre important de lignes de code, cela indique qu'un effort relativement important a été fourni pour écrire cette classe de test [Bruntink 04, Bruntink 06].
- TAssert : cette métrique donne le nombre d'occurrences textuelles d'Assert JUnit se produisant dans le code d'une classe de test. Les testeurs se servent de l'ensemble des méthodes assert JUnit pour comparer le comportement attendu de la classe sur laquelle est appliqué un test JUnit et son comportement réel (pas sous l'application d'un test) : cela permet de donner une idée sur la taille d'une suite de tests et la construction des cas de tests [Bruntink 04, Bruntink 06].

- TInvok : cette métrique renseigne sur le nombre de méthodes invoquées par une classe de test : il s'agit des méthodes utilisées par les testeurs pour supporter les interactions entre les classes en cours de test et les classes avec qui elles collaborent. TInvok est utilisée pour indiquer l'effort requis pour tester les méthodes d'une classe couplée à d'autres classes.
- TData : lorsque des tests JUnit sont exécutés, il peut arriver que de nouveaux objets soient créés en vue de l'utilisation des attributs et des méthodes d'une classe donnée. La métrique TData donne une mesure du nombre de ces objets créés lors des tests.

Les deux dernières métriques ont été développées dans le cadre de travaux connexes au laboratoire de recherche en génie logiciel du Département de mathématiques et d'informatique de L'UQTR. Ainsi, nous nous servons des données de ces quatre métriques de test pour quantifier et mesurer la testabilité (en termes de caractéristiques des classes test) lors de nos expérimentations qui concernent entre autres l'effort de test. Il est clair que dans nos travaux, la testabilité est abordée sous la perspective des tests unitaires des classes et plus particulièrement sous l'angle de l'effort de l'écriture du code des tests JUnit correspondant aux classes testées.

2.3. Méthodes d'expérimentation

2.3.1 Corrélations

En probabilités et en statistique, étudier la corrélation entre deux ou plusieurs variables aléatoires ou statistiques numériques, c'est étudier l'intensité de la liaison qui peut exister entre ces variables. Dans nos expérimentations, nous nous servons des corrélations pour faire une analyse des informations contenues dans les différentes métriques et observer les liens pouvant exister entre elles. Afin de consolider nos hypothèses, nous utiliserons deux (02) types de corrélations souvent utilisées dans la littérature, à savoir, la corrélation de Pearson et la corrélation de Spearman [Bonett 00]. Le seuil de significativité retenu est 5%.

2.3.1.1. Corrélation de Pearson

La corrélation de Pearson est une mesure de la corrélation, ou d'une dépendance linéaire, entre deux variables X et Y , donnant une valeur comprise en $+1$ et -1 inclusivement. Une valeur de 1 signifie qu'une équation linéaire décrit parfaitement la relation entre X et Y , avec des points de données situés sur une ligne tels que si les valeurs de Y augmentent alors les valeurs de X augmentent. Une valeur de -1 peut être illustrée avec des points de données situés sur une ligne tels que si les valeurs de Y diminuent alors les valeurs de X augmentent et vice versa. Si le coefficient de corrélation vaut 0 , alors il n'y a aucune dépendance linéaire entre les variables [Good 09].

2.3.1.2. Corrélation de Spearman

Cette technique, basée sur les rangs d'observation, est largement utilisée pour mesurer le degré de relation linéaire entre deux variables. Elle consiste à trouver un coefficient de corrélation, non pas entre les valeurs prises par les deux variables mais entre les rangs de ces valeurs. Le coefficient de corrélation de Spearman prend une valeur comprise entre -1 et +1. Analogiquement à la corrélation de Pearson, une corrélation positive est une corrélation dans laquelle les rangs des deux variables augmentent en même temps. Une corrélation négative est une corrélation dans laquelle les rangs d'une variable augmentent tandis que ceux de l'autre variable diminuent. Une corrélation entre deux variables sera de +1 ou -1 si la relation entre les rangs est exactement linéaire. Une corrélation proche de zéro signifie qu'il n'y a pas de relation linéaire entre les rangs.

2.3.2 Régression logistique

La régression logistique est utilisée pour la prédiction d'une variable binaire à partir d'un ensemble de variables indépendantes pour déterminer le pourcentage de variance de la variable dépendante expliquée par les variables indépendantes (dites variables explicatives) [Dreiseitl 02]. La régression logistique est de deux types : univariée et multivariée. La régression logistique univariée est une méthode statistique utilisée pour générer des probabilités à partir d'un ensemble de variables aléatoires. Cette technique permet de mieux cerner la contribution d'une variable indépendante particulière dans la prédiction d'une variable binaire et construire un modèle de prédiction à partir des observations retenues. La régression logistique multivariée quant à elle, est

essentiellement utilisée pour la construction d'un modèle de prédiction en combinant plusieurs variables dites explicatives.

2.3.3 K-Means

L'algorithme des k-moyennes ou K-Means est un algorithme de partitionnement de données, c'est-à-dire une méthode dont le but est de diviser des observations en clusters (catégories/classes) dans lesquels chaque observation appartient à la partition avec la moyenne la plus proche. Les nuées dynamiques sont une généralisation de ce principe, pour laquelle chaque partition est représentée par un noyau pouvant être plus complexe qu'une moyenne.

K-Means nous permettra de déterminer les caractéristiques des classes par rapport aux valeurs des données de leurs métriques.

2.4. Outils d'expérimentation

Pour mener à bien nos multiples expérimentations, nous avons eu recours à trois principaux outils implémentant des algorithmes :

- de calculs des valeurs des métriques orientées-objet des systèmes logiciels sélectionnés, de méthodes statistiques (régressions linéaire et logistique, corrélations, etc.) ;
- de méthodes utilisées dans le domaine de l'intelligence artificielle (arbres de décision) ;

- et de classification (K-Means).

2.4.1 Borland Together

Together [Together] est un ensemble de plugins Eclipse écrits en Java. Parmi ses nombreuses fonctionnalités, nous avons exploité les calculs automatiques des valeurs des métriques orientées-objet, que ce soit au niveau des classes logicielles ou des classes de test.

2.4.2 XLSTAT

XLSTAT [Xlstat] est un add-on de statistiques et d'analyses de données pour Microsoft Excel. Il offre de nombreuses fonctionnalités et des utilitaires pour faciliter la manipulation de données sous Excel. XLSTAT permet notamment de réaliser des analyses (ACP, corrélations, etc.), diverses méthodes de régression et des tests (régression linéaire et logistique).

2.4.3 Weka

Weka [Weka] (Waikato Environment for Knowledge Analysis) est une suite populaire de logiciels d'apprentissage automatique écrite en Java et sous licence gratuite (GNU GPL).

On y retrouve, entre autres, une implémentation de la méthode de classification K-Means.

CHAPITRE 3. CATÉGORISATION ET TAUX DE COUVERTURE

Cette partie consistera en une description du taux de couverture, notion qui sera abordée tout au long de nos expérimentations. Par ailleurs, une catégorisation des classes orientées-objet va nous permettre de segmenter les données dans le but d'affiner les diverses observations que nous ferons par rapport aux métriques orientés-objet, aux taux de couverture, à l'effort de test, etc.

3.1. Taux de couverture

3.1.1 Définition

Le taux de couverture des tests est une mesure de la proportion exécutée d'un programme lors d'une série de tests [Mockus 09]. Autrement dit, elle permet de mesurer les parties du code source (testé) qui sont couvertes par les tests. Ceci permet de déterminer les parties qui ne sont pas couvertes, donc de développer des tests supplémentaires. Étant un taux, il s'exprime donc en pourcentage et informe sur les parties d'un programme exécutées pendant les tests.

L'outil Together [Together] fournit la valeur de la métrique JUC (JUnit Test Coverage) que nous noterons Tcov dans le cadre de cette étude : il s'agit d'une valeur associée à chaque classe constituant un système donné ; si elle est nulle, cela signifie qu'aucune méthode de la classe en question n'a été appelée lors de l'exécution des tests JUnit. Si toutefois, elle est non nulle, cela signifie qu'au

moins une méthode de la classe a été appelée au moins une fois par une autre classe lors de l'exécution des tests unitaires.

Soit M le nombre de méthodes d'une classe C donnée et N le nombre de méthodes appelées au moins une fois à l'exécution des tests :

$$Tcov(C) = (N / M) * 100.$$

Notons qu'une classe peut par conséquent avoir une valeur non nulle de son taux de couverture bien qu'aucun test JUnit n'ait été explicitement développé pour celle-ci : en effet, elle peut tout à fait être appelée (une ou des méthodes la constituant peuvent être appelées) par une autre classe pour laquelle des tests JUnit ont été développés (explicitement).

3.1.2 Limitation

Voici un aperçu des valeurs du taux de couverture $Tcov$ pour quelques classes du système JFREECHART :

Resource	Tcov	NOO
org.jfree.chart.axis.SegmentedTimeline	76	43
org.jfree.chart.labels.StandardCategoryItemLabelGenerator	100	1
org.jfree.data.gantt.Task	100	13
org.jfree.data.KeyedObject	100	5
org.jfree.data.statistics.Regression	100	4
org.jfree.chart.annotations.CategoryTextAnnotation	90	9
org.jfree.chart.labels.StandardXYToolTipGenerator	88	4

Tableau 1 - Échantillon de valeurs du taux de couverture

Nous pouvons constater que la classe `SegmentedTimeline` compte 43 opérations (NOO) et son Tcov vaut 76% : cela signifie que 33 opérations sur les 43 qu'elle compte, sont appelées au moins une fois lors de l'exécution des tests JUnit.

Par ailleurs, si nous nous intéressons à la classe `StandardCategoryItemLabelGenerator`, nous remarquons que celle-ci ne compte qu'une (01) opération qui est appelée au moins une fois lors de l'exécution des tests, ce qui lui vaut une valeur de Tcov de 100%.

Il ressort de ces deux cas, qui ne sont assurément pas des cas isolés, qu'il devient difficile, voire impossible d'effectuer une comparaison des valeurs brutes de la métrique Tcov par classe, étant donné que celles-ci ne reflètent que le ratio de méthodes appelées au moins une fois lors de l'exécution des tests JUnit et ne permettent pas d'avoir un visu global sur l'étendue de la couverture des tests sur l'ensemble d'un système donné, ou du moins, d'en avoir une idée générale (globale).

Par conséquent, si nous désirons mener une étude expérimentale basée sur le taux de couverture, il apparaît comme plus pertinent (voire indispensable) de reconsidérer le calcul de cette métrique par l'outil Together afin de permettre des analyses plus fiables (objectives) et basées sur des valeurs prenant en compte l'ensemble des classes d'un système donné, et non basées sur la seule donnée d'une classe par rapport à elle-même.

3.1.3 Re-calcul

Suite au précédent constat tiré du *tableau 1* et dans le but de pouvoir faire une analyse comparative objective des valeurs de Tcov par classe et avoir un visu plus étendu de la couverture des tests JUnit sur l'ensemble d'un système donné, nous nous sommes alors proposé d'affecter une pondération à chaque valeur de Tcov.

Sachant que Tcov se calcule à partir du nombre d'opérations N appelées d'une classe C sur le nombre total d'opérations M, le coefficient de pondération désigné ne serait autre que la métrique NOO.

Soit WTcov(C) (*Weighted Test Coverage*) le taux de couverture pondéré d'une classe C :

$$WTcov(C) = NOO(C) * Tcov(C) = NOO(C) * (N / M) * 100.$$

Équation 1 - Détermination du taux de couverture pondéré

En reprenant les valeurs du *tableau 1* précédent, nous obtenons :

Resource	Tcov	NOO	WTcov
org.jfree.chart.axis.SegmentedTimeline	76	43	3268
org.jfree.chart.labels.StandardCategoryItemLabelGenerator	100	1	100
org.jfree.data.gantt.Task	100	13	1300
org.jfree.data.KeyedObject	100	5	500
org.jfree.data.statistics.Regression	100	4	400
org.jfree.chart.annotations.CategoryTextAnnotation	90	9	810
org.jfree.chart.labels.StandardXYToolTipGenerator	88	4	352

Tableau 2 - Données du tableau 1 après re-calcul du taux de couverture

Comme la valeur du taux de couverture (WTcov) est pondérée par le nombre d'opérations, nous sommes en mesure de mieux discerner la portée réelle de la couverture des tests JUnit : en effet, en regardant les 2 premières lignes du *tableau 2*, nous constatons que la classe *<SegmentedTimeline>* contient 43 opérations et environ 33 (Tcov = 76%) d'entre elles sont appelées lors de l'exécution des tests JUnit, tandis que la classe *<StandardCategoryItemLabelGenerator>* ne contient qu'une seule opération, et c'est celle-ci qui est appelée lors des tests JUnit.

La pondération avec le nombre d'opérations (métrique NOO) que compte une classe trouve tout son sens par le fait que si nous nous référons aux données du *tableau 1* les taux de couverture obtenus pour les deuxième, quatrième et cinquième classes de notre tableau, nous avons un taux de couverture de 100%, ce qui représente en soi une valeur significative. Toutefois, ces 3 classes possèdent relativement peu de méthodes (NOO entre 1 et 5, totalisant 10 méthodes). D'un autre côté, la première classe du tableau, dont le taux de couverture est de 76%, compte 43 méthodes, et 33 méthodes sur ces 43 sont toutes appelées lors de l'exécution des tests. En considérant les taux de couverture de 100% et de 76%, nous ne pouvons pas réellement cerner combien de méthodes sont concernées et nous ne pouvons pas non plus comparer les classes entre elles.

Grâce à notre nouveau mode de calcul du taux de couverture, nous obtenons un taux de couverture pondéré de 3268 pour la première classe (le taux de couverture initial étant de 76%, soit 33 méthodes) et pour les deuxième, quatrième et cinquième classes, initialement à 100% (totalisant 10 méthodes), nous obtenons respectivement un taux de couverture pondéré de 100, 400 et

500. Ces dernières données traduisent bien l'impact réel de la couverture des tests de chacune des classes par rapport à l'ensemble d'un système donné.

Ainsi, le taux de couverture pondéré obtenu, comme le montre les données du *tableau 2*, offre plus de transparence quant à la compréhension et la comparaison logique des taux de couverture des classes.

3.2. Catégorisation des classes orientées-objet

Rappels :

- Nos expérimentations seront menées sur quatre systèmes développés en Java, à savoir : JFREECHART, JODA, IO, POI;
- Pour chacune des classes, de chacun de ces systèmes, nous disposons des valeurs des métriques orientées-objet, obtenues grâce à l'outil Together;
- Des tests JUnit ont été développés pour quelques classes sur chaque système et par conséquent, nous disposons également des valeurs des métriques de test orientées-objet pour ces classes;
- Nos expérimentations se basent essentiellement sur le taux de couverture, dont la formule de calcul a été repensée pour refléter plus d'objectivité et permettre d'étendre l'analyse des valeurs sur l'ensemble des classes d'un système donné.

Ainsi, pour chaque système nous avons distingué trois catégories de classes :

- CAT-2 : classes pour lesquelles des tests JUnit ont été développés et pour lesquelles nous avons des données de taux de couverture,
- CAT-1 : classes pour lesquelles nous avons des données de taux de couverture alors que nous n'avons pas de tests JUnit développés pour celles-ci. Il s'agit ici des classes dont les méthodes sont appelées par d'autres classes (pour lesquelles des tests JUnit ont été développés) à l'exécution des tests JUnit.
- CAT-0 : classes dont nous n'avons pas de données de taux de couverture et pour lesquelles aucun test JUnit n'a été développé.

Le *tableau 3* offre un récapitulatif de la répartition des trois différentes catégories pour chaque système :

	JFREECHART	JODA	IO	POI
CAT-2	223 (44,95 %)	75 (33,33 %)	59 (54,12 %)	359 (23,33 %)
CAT-1	87 (17,55 %)	96 (42,67 %)	25 (22,94 %)	574 (37,30 %)
CAT-0	186 (37,50 %)	54 (24 %)	25 (22,94 %)	606 (39,37 %)

Tableau 3 - Répartition des 3 différentes catégories

Nous remarquons que la répartition des catégories sur l'ensemble des quatre systèmes est vraiment hétérogène, ce qui constitue un environnement expérimental idéal dans le cadre de notre étude, tel que le montre le *tableau 3*.

CHAPITRE 4. OBSERVATION DES MÉTRIQUES ORIENTÉES-OBJET VERSUS LA COUVERTURE DES TESTS

4.1. Objectif

Ce chapitre a pour but de dégager, à partir d'une analyse approfondie portant sur les données des métriques de systèmes orientés-objet, des éléments caractéristiques frappants permettant de dresser si possible un *pattern* (ou patron) établissant un lien entre les valeurs des métriques orientées-objet obtenues à partir du code source des systèmes JFREECHART, JODA et IO et le taux de couverture des tests JUnit, obtenu à partir du code source des jeux de tests correspondants.

4.2. Hypothèse

En se basant sur l'objectif de ce chapitre, énoncé précédemment, notre hypothèse serait donc qu'il existerait un lien entre les métriques orientées-objet et le taux de couverture qui permettrait de faire ressortir un *pattern* qui caractériserait un système orienté-objet donné, au niveau de la couverture des tests effectués sur celui-ci.

Notre démarche expérimentale consistera en quatre étapes distinctes, développées dans ce qui suit : en effet, nous nous préoccuperons avant tout des classes pour lesquelles nous observons des valeurs non nulles du taux de couverture, en d'autres termes, des classes des catégories CAT-2 et CAT-1; ceci dit, une attention particulière doit également être portée aux classes de la catégorie CAT-0, car bien que les valeurs de la métrique de couverture des tests

JUnit soient nulles, il est probable que des caractéristiques précises de ces classes aient fait que les développeurs n'ont pas développé explicitement des tests JUnit pour celles-ci :

- Procéder à une analyse des statistiques descriptives des données de métriques de chacun des quatre systèmes que nous avons sélectionnés, pour chacune des catégories CAT-2, CAT-1 et CAT-0. Cela nous offrira un aperçu global des valeurs des données et nous permettra de dégager quelques éléments caractéristiques des systèmes;
- Faire une analyse de corrélations (Spearman et Pearson) entre les métriques orientées-objet et le taux de couverture, pour chacune des catégories CAT-2 et CAT-1. La significativité ou non de la corrélation entre une métrique donnée et le taux de couverture sera un élément intéressant pour faire ressortir une caractéristique du système étudié;
- Faire une analyse graphique des tendances qui se dégagent de la comparaison des valeurs des métriques orientées-objet et celles du taux de couverture pour chacune des catégories CAT-2 et CAT-1;
- Appliquer une classification K-Means (avec $k = 3$ = le nombre de catégories sur lesquelles porte notre expérimentation) sur les données des métriques des classes des catégories CAT-2, CAT-1 et CAT-0: de ce fait, nous tâcherons de faire ressortir des caractéristiques particulières supplémentaires de chaque catégorie.

4.3. Systèmes logiciels

Nous avons sélectionné quatre systèmes qui sont soit des API, soit des packages écrits en Java, et dont nous disposons des codes sources (open source) pour calculer les valeurs des métriques orientées-objet. Parmi ces quatre systèmes, trois sont retrouvés dans la littérature pour des recherches connexes aux nôtres, à savoir JFREECHART, JODA et IO [Kaur 13], et le quatrième, POI, a été choisi en raison du nombre important de classes qu'il possède (POI : 1539) et du nombre conséquent des lignes de code qu'il compte (POI : 132445). Nous avons, en fait, analysé d'autres systèmes. Mais, pour des raisons d'espace, nous nous limitons dans ce mémoire à présenter uniquement les quatre. Les résultats obtenus dans le cadre des autres systèmes (non présentés dans ce mémoire), suivent en gros les mêmes tendances des systèmes présentés.

Les quatre systèmes se différencient les uns des autres par les équipes de développeurs qui les ont réalisés et par les fonctionnalités qu'ils intègrent. Ils diffèrent aussi en termes de taille et de complexité.

4.3.1 JFREECHART

JFREECHART [Jfreechart] est une bibliothèque graphique gratuite (LGPL) pour la plate-forme Java. Il supporte les graphiques à barres, en camemberts, en courbes, les séries chronologiques, les diagrammes de dispersion, les histogrammes, les diagrammes de Gantt, les diagrammes de Pareto, les parcelles de bulles, les cadrans, les thermomètres, etc.

4.3.2 JODA

JODA [Joda] est une librairie Java qui permet de rendre les valeurs de type date/heure plus faciles à gérer, à manipuler et à comprendre. Divers systèmes de calendriers sont fournis par l'API. Par défaut, le calendrier disponible suit le standard ISO8601 qui est utilisé par XML. On y retrouve entre autres le calendrier grégorien, julien, bouddhiste et islamique. Les classes intègrent aussi la gestion du fuseau horaire, de la durée, du formatage et de la conversion de la date ou de l'heure.

4.3.3 IO

IO [Io] est un package Java supportant les opérations d'input et d'output et fournit toutes les classes nécessaires à la création, lecture, écriture et traitement des flux tels que:

- les flux de caractères et les flux binaires;
- les flux de communication et les flux de traitement;
- les flux avec ou sans tampon;
- ...

Une classe permettant un accès aléatoire aux fichiers est aussi disponible.

Le package intègre également diverses classes pour soutenir les interactions avec le système de fichiers hôte.

4.3.4 POI

POI [Poi] est un projet de l'Apache Software Foundation permettant de manipuler avec la technologie Java divers types de fichiers créés par Microsoft Office comme Word, PowerPoint et Excel.

En d'autres termes, POI offre la possibilité de créer, de lire et de modifier des fichiers MS Excel, MS Word et MS Powerpoint en Java.

4.4. Expérimentation

Pour mener à bien notre étude et afin de vérifier la validité de notre hypothèse, nous suivrons les étapes de la démarche expérimentale citée précédemment une à une pour chaque catégorie de classes orientées-objet et pour chacun des 4 systèmes.

À cet effet, nous commenterons chaque résultat dans la partie « discussion » et émettrons une « conclusion » sur l'expérimentation.

4.4.1 JFREECHART

4.4.1.1. Analyse des statistiques descriptives

Tel qu'énoncé dans la description de la démarche expérimentale, la première étape consiste en une analyse des statistiques descriptives des données de métriques de chaque système, pour chacune des catégories CAT-2, CAT-1 et CAT-0. Cela va permettre de dégager quelques éléments caractéristiques de chaque système.

Les tableaux correspondants aux statistiques descriptives par catégorie de classes sont représentés ci-contre :

Variable	Observation	Minimum	Maximum	Moyenne
CBO	223	0,000	67,000	16,193
COC	223	0,000	101,000	6,578
FO	223	0,000	49,000	11,054
DOIH	223	1,000	5,000	2,350
NOCC	223	0,000	50,000	1,130
LCOM	217	0,000	13524,000	354,018
LOC	223	21,000	2041,000	237,753
NOA	223	0,000	47,000	5,018
NOO	223	1,000	170,000	17,852
RFC	223	15,000	671,000	152,466
WMPC	223	3,000	470,000	47,251
WTcov	223	20,000	11550,000	765,381

Tableau 4 - Statistiques descriptives des données des classes de CAT-2

Variable	Observation	Minimum	Maximum	Moyenne
CBO	87	0,000	101,000	4,943
COC	87	0,000	55,000	6,713
FO	87	0,000	56,000	3,483
DOIH	87	0,000	5,000	1,138
NOCC	87	0,000	66,000	8,333
LCOM	39	0,000	297,000	30,333
LOC	87	3,000	1044,000	55,057
NOA	87	0,000	12,000	1,046
NOO	87	0,000	132,000	8,402
RFC	87	1,000	301,000	29,851
WMPC	87	1,000	132,000	14,069
WTcov	87	0,000	2006,000	270,115

Tableau 5 - Statistiques descriptives des données des classes de CAT-1

Variable	Observation	Minimum	Maximum	Moyenne
CBO	186	0,000	38,000	4,914
COC	186	0,000	24,000	1,527
FO	186	0,000	26,000	3,339
DOIH	186	0,000	7,000	1,882
NOCC	186	0,000	25,000	0,817
LCOM	108	0,000	551,000	32,750
LOC	186	2,000	385,000	56,468
NOA	186	0,000	19,000	1,828
NOO	186	0,000	38,000	5,651
RFC	186	0,000	560,000	78,419
WMPC	186	0,000	86,000	11,382

Tableau 6 - Statistiques descriptives des données des classes de CAT-0

L'analyse des *tableaux 4, 5 et 6* fait ressortir un certain nombre de points à savoir:

- JFREECHART compte 496 classes dont près de la moitié (45%) appartiennent à la catégorie CAT-2, 17,50% appartiennent à la catégorie CAT-1 et 37,50% à CAT-0: déjà, avec la proportion des classes de CAT-0, nous pouvons noter qu'une partie non négligeable de

ce système n'a pas été testée (pas de tests JUnit développés) et n'est pas atteinte (couverte) par les tests des autres classes (par appel);

- Les classes de la catégorie CAT-2 (*tableau 4*) présentent des valeurs relativement élevées pour ce qui est de la complexité (WMPC et RFC: 47 et 152 en moyenne), la taille (LOC, NOA et NOO), le couplage sortant (CBO, FO) et la cohésion. Rappelons qu'il s'agit ici des classes pour lesquelles des tests JUnit ont été développés et dont bien entendu nous disposons de valeurs du taux de couverture pondéré non nulles voire même relativement élevées (moyenne: 765,38). En partant de la formule de calcul du taux de couverture pondéré (cf. *Chapitre 3*) et en notant que le nombre d'opérations (métrique NOO) en moyenne pour chaque classe de la CAT-2 est de 17,85, on peut avancer qu'en moyenne 43% ($765,38 / 17,85$) des méthodes de chaque classe est appelée lors des tests, ce qui est vraiment pertinent car cela représente près de la moitié des méthodes d'une classe qui est appelée par d'autres classes lors des tests: cela nous amène à porter un intérêt particulier aux métriques de couplage;
- Ce qui serait le plus notable quant aux classes de la catégorie CAT-1 (*tableau 5*), ce serait le fait que le couplage entrant (FAN-In ou COC) est comparativement plus important que le couplage sortant (Fan-Out): la moyenne des valeurs du FAN-In est de 6,71 contre 3,48 pour celle de Fan-Out. Cela pourrait s'expliquer par le fait que ces classes sont appelées lors des tests menés sur d'autres classes (appelantes) et de ce fait la couverture des tests est plus importante: en effet, les classes de la catégorie CAT-1, seraient en quelque sorte également testées, sans

l'être explicitement (car aucun test JUnit n'a été développé pour ces classes).

- En se basant sur les observations du *tableau 6*, force est de constater que les classes de la catégorie CAT-0 présentent un couplage (CBO, FO, COC) relativement faible (entre 1,5 et 4,9), en plus de valeurs des métriques de taille (LOC, NOA, NOO) toutes aussi relativement faibles comparativement aux valeurs de ces trois métriques de taille pour les classes des catégories CAT-2 et CAT-1. Ces classes n'ont pas été testées et ne sont pas atteintes par les tests d'autres classes bien qu'elles représentent 37,5% des classes que compte le système.
- Notons enfin que l'héritage semble ne pas être une caractéristique déterminante liée au taux de couverture étant donné que les moyennes sont sensiblement les mêmes dans les 3 catégories.

Il apparaît assez distinctement que les classes des catégories CAT-2 et CAT-0 s'opposent du point de vue de leurs caractéristiques. Tandis que celles de CAT-2 ont des valeurs élevées des métriques de taille, de complexité et de couplage (nous laisserons de côté les métriques de cohésion du fait du manque de données non négligeable), celles de CAT-0 ont des valeurs très faibles des données des mêmes métriques.

Les classes de CAT-1 ont un Fan-In (COC) notable par rapport à leur Fan-Out. Étant atteintes lors des tests des classes de CAT-2, cela pourrait expliquer qu'elles présentent des valeurs de taux couverture non nulles bien que n'ayant pas de tests JUnit explicitement développés pour elles.

Ainsi, les familles de métriques à suivre de près seraient la taille, la complexité et le couplage: les métriques de cohésion étant biaisées par le fait qu'il nous manque beaucoup de données et les métriques d'héritage ne semblant pas être une caractéristique particulière distinguant les classes entre elles. Les expérimentations sur les systèmes à suivre nous éclaireront plus à ce sujet, ainsi que la réalisation des prochaines étapes de notre démarche expérimentale.

4.4.1.2. Analyse des corrélations (Spearman et Pearson)

L'analyse de corrélations de Spearman et Pearson entre les métriques orientées-objet et le taux de couverture, pour chacune des catégories CAT-2 et CAT-1 servira à déterminer la significativité ou non de la corrélation entre une métrique donnée et le taux de couverture afin de faire ressortir une ou des caractéristiques du système.

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	-0,053	0,982	0,572	0,271	0,646	0,795	0,509	0,554	0,859	0,699	0,303
COC	-0,053	1	-0,053	-0,374	0,439	0,183	0,163	0,273	0,381	-0,146	0,274	0,529
FO	0,982	-0,053	1	0,581	0,294	0,626	0,784	0,474	0,535	0,872	0,683	0,289
DOIH	0,572	-0,374	0,581	1	0,065	0,339	0,370	0,029	0,200	0,761	0,283	-0,030
NOCC	0,271	0,439	0,294	0,065	1	0,334	0,291	0,278	0,311	0,264	0,294	0,261
LCOM	0,646	0,183	0,626	0,339	0,334	1	0,738	0,738	0,664	0,755	0,541	0,752
LOC	0,795	0,163	0,784	0,370	0,291	0,738	1	0,682	0,825	0,736	0,966	0,543
NOA	0,509	0,273	0,474	0,029	0,278	0,664	0,682	1	0,776	0,311	0,730	0,570
NOO	0,554	0,381	0,535	0,200	0,311	0,755	0,825	0,776	1	0,500	0,907	0,782
RFC	0,859	-0,146	0,872	0,761	0,264	0,541	0,736	0,311	0,500	1	0,634	0,212
WMPC	0,699	0,274	0,683	0,283	0,294	0,752	0,966	0,730	0,907	0,634	1	0,655
WTcov	0,303	0,529	0,289	-0,030	0,261	0,501	0,543	0,570	0,782	0,212	0,655	1

Tableau 7 - Corrélations de Spearman des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,123	0,962	0,087	-0,023	0,616	0,679	0,418	0,629	0,511	0,705	0,327
COC	0,123	1	0,153	-0,505	0,556	0,218	0,090	-0,016	0,485	-0,182	0,295	0,753
FO	0,962	0,153	1	0,018	0,032	0,618	0,592	0,313	0,582	0,425	0,620	0,355
DOIH	0,087	-0,505	0,018	1	-0,650	-0,223	0,495	0,485	-0,072	0,686	0,221	-0,512
NOCC	-0,023	0,556	0,032	-0,650	1	0,349	-0,265	-0,342	0,176	-0,349	-0,055	0,489
LCOM	0,616	0,218	0,618	-0,223	0,349	1	0,674	0,312	0,658	0,626	0,941	0,505
LOC	0,679	0,090	0,592	0,495	-0,265	0,674	1	0,665	0,766	0,821	0,565	0,259
NOA	0,418	-0,016	0,313	0,485	-0,342	0,312	0,665	1	0,436	0,523	0,669	0,009
NOO	0,629	0,485	0,582	-0,072	0,176	0,658	0,766	0,436	1	0,463	0,910	0,668
RFC	0,511	-0,182	0,425	0,686	-0,349	0,626	0,821	0,523	0,463	1	0,669	-0,053
WMPC	0,705	0,295	0,620	0,221	-0,055	0,941	0,565	0,910	0,669	0,669	1	0,475
WTcov	0,327	0,753	0,355	-0,512	0,489	0,505	0,259	0,009	0,668	-0,053	0,475	1

Tableau 8 - Corrélations de Spearman des métriques OO des classes de CAT-1

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,222	0,985	0,460	0,216	0,503	0,800	0,663	0,675	0,765	0,727	0,433
COC	0,222	1	0,249	-0,170	0,543	0,474	0,420	0,376	0,510	0,032	0,489	0,571
FO	0,985	0,249	1	0,454	0,236	0,524	0,814	0,660	0,684	0,786	0,742	0,440
DOIH	0,460	-0,170	0,454	1	-0,098	0,066	0,269	0,085	0,163	0,690	0,199	-0,020
NOCC	0,216	0,543	0,236	-0,098	1	0,369	0,308	0,370	0,420	0,115	0,351	0,521
LCOM	0,503	0,474	0,524	0,066	0,369	1	0,810	0,809	0,889	0,279	0,891	0,619
LOC	0,800	0,420	0,814	0,269	0,308	0,810	1	0,836	0,914	0,557	0,969	0,720
NOA	0,663	0,376	0,660	0,085	0,370	0,809	0,836	1	0,907	0,364	0,855	0,725
NOO	0,675	0,510	0,684	0,163	0,420	0,889	0,914	0,907	1	0,379	0,962	0,861
RFC	0,765	0,032	0,786	0,690	0,115	0,279	0,557	0,364	0,379	1	0,456	0,171
WMPC	0,727	0,489	0,742	0,199	0,351	0,891	0,969	0,855	0,962	0,456	1	0,811
WTcov	0,433	0,571	0,440	-0,020	0,521	0,619	0,720	0,725	0,861	0,171	0,811	1

Tableau 9 - Corrélations de Pearson des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,238	0,976	0,055	-0,060	0,405	0,962	0,186	0,407	0,708	0,763	0,554
COC	0,238	1	0,254	-0,275	0,189	0,137	0,239	0,063	0,181	0,027	0,225	0,469
FO	0,976	0,254	1	0,015	0,002	0,432	0,922	0,161	0,561	0,736	0,845	0,596
DOIH	0,055	-0,275	0,015	1	-0,362	-0,170	0,112	0,153	-0,124	0,390	-0,003	-0,229
NOCC	-0,060	0,189	0,002	-0,362	1	0,089	-0,121	-0,203	0,141	-0,122	0,010	0,071
LCOM	0,405	0,137	0,432	-0,170	0,089	1	0,619	0,598	0,879	0,283	0,728	0,756
LOC	0,962	0,239	0,922	0,112	-0,121	0,619	1	0,263	0,390	0,734	0,783	0,613
NOA	0,186	0,063	0,161	0,153	-0,203	0,598	0,263	1	0,098	0,274	0,266	0,245
NOO	0,407	0,181	0,561	-0,124	0,141	0,879	0,390	0,098	1	0,519	0,867	0,595
RFC	0,708	0,027	0,736	0,390	-0,122	0,283	0,734	0,274	0,519	1	0,755	0,338
WMPC	0,763	0,225	0,845	-0,003	0,010	0,728	0,783	0,266	0,867	0,755	1	0,690
WTcov	0,554	0,469	0,596	-0,229	0,071	0,756	0,613	0,245	0,595	0,338	0,690	1

Tableau 10 - Corrélations de Pearson des métriques OO des classes de CAT-1

Après avoir calculé les corrélations de Spearman et de Pearson des métriques orientées-objet et du taux de couverture à l'aide de l'outil XLSTAT, passons à présent à l'analyse respective des *tableaux 7, 8, 9 et 10* :

- Les tableaux de corrélations révèlent bien à quel point la corrélation entre le FAN-In et le taux de couverture est significative pour la catégorie CAT-1 (0,75 versus 0,53 pour CAT-2: corrélation de Spearman, *tableaux 7 et 8*).
- Dans les 2 catégories, nous voyons que l'héritage (à travers DOIH) n'est pas corrélé au taux de couverture.
- La cohésion (LCOM) également est fortement corrélée au taux de couverture avec une plus forte intensité en faveur de CAT-2.

- La complexité quant à elle, via la métrique WMPC, est significativement corrélée au taux de couverture dans les 2 catégories (CAT-2: 0,65 et CAT-1: 0,48 pour Spearman).
- Il est tout à fait normal que la corrélation entre la métrique NOO et le taux de couverture soit forte ($> 0,65$) car le calcul du taux de couverture s'est fait à partir des valeurs de NOO.
- Notons également que la métrique NOA n'a aucune corrélation avec le taux de couverture, et cela peut se constater dans les statistiques descriptives du *tableau 5*, où la moyenne des valeurs de NOA est relativement plus faible dans la catégorie CAT-1 comparée à celle trouvée dans la catégorie CAT-2 (1,05 contre 5,02, *tableau 4*). Ce facteur aurait pu être déterminant quant au fait qu'une classe soit testée ou pas. Si nous nous penchons sur la métrique LOC, nous voyons que, bien que cette métrique soit significativement corrélée au taux de couverture dans les 2 catégories, il y a une nette différence quant à l'intensité de la corrélation (0,26 pour CAT-1 et 0,54 pour CAT-2 pour Spearman, c'est ce que révèlent les données des *tableaux 7 et 8*) : la métrique LOC pourrait donc aussi être déterminante dans la décision de tester une classe ou pas.

L'analyse des corrélations révèle un certain nombre d'informations supplémentaires quant aux caractéristiques des classes pour lesquelles des tests JUnit ont été développés, mais également pour celles pour lesquelles nous avons des données de métrique du taux de couverture non nulles alors qu'il n'y a pas eu de tests développés pour ces classes: en effet, les classes de la catégorie

CAT-1 possèdent un FAN-In (COC) fortement corrélé au taux de couverture (voir le *tableau 8*), tandis que les classes de CAT-2 se démarquent surtout par les métriques LOC et NOA, plus corrélées au taux de couverture que celles de CAT-1 (*tableaux 7 et 9*). Ces caractéristiques de taille pourraient être des facteurs dans la détermination des tests à développer.

Une caractéristique commune aux deux catégories, serait la forte corrélation positive de la métrique de complexité WMPC ainsi que de la métrique de cohésion LCOM avec le taux de couverture.

4.4.1.3. Analyse des courbes de tendances

En sélectionnant quelques métriques (COC, FO, DOIH, LCOM, LOC et WMPC) à la suite de l'étape expérimentale précédente, tâchons d'analyser les tendances de variation des valeurs des métriques et du taux de couverture dans chaque catégorie.

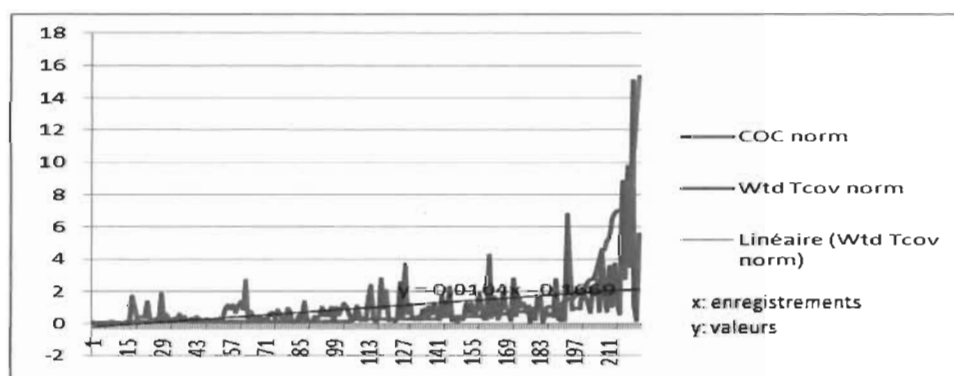


Figure 1 - Variation du taux de couverture vs la métrique COC (CAT-2)

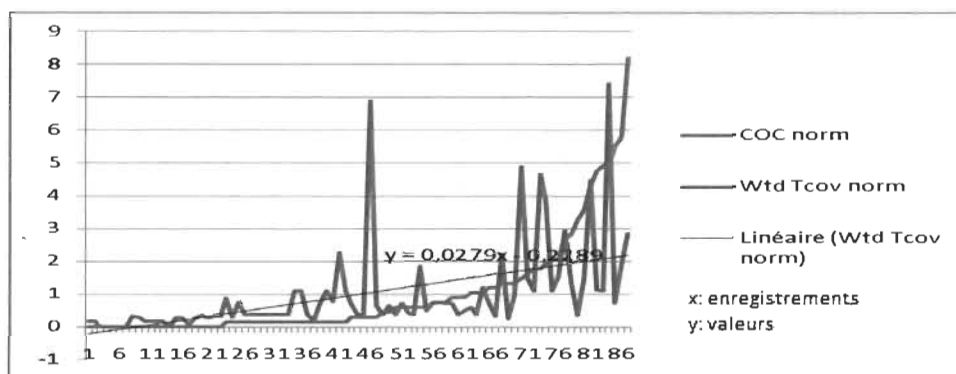


Figure 2 - Variation du taux de couverture vs la métrique COC (CAT-1)

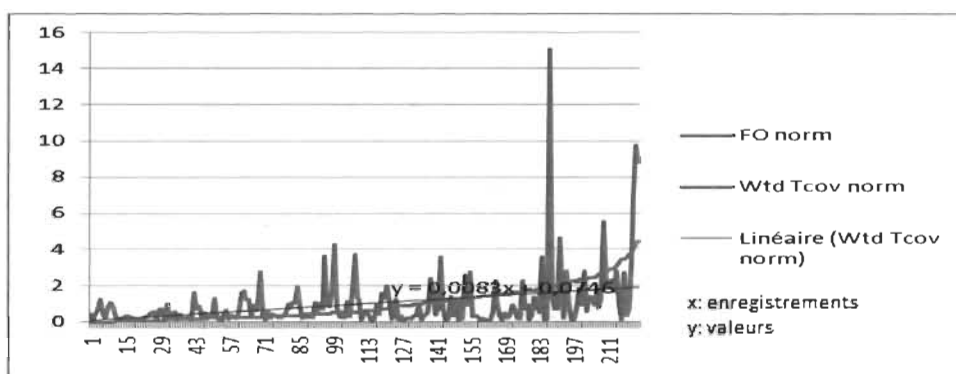


Figure 3 - Variation du taux de couverture vs la métrique FO (CAT-2)

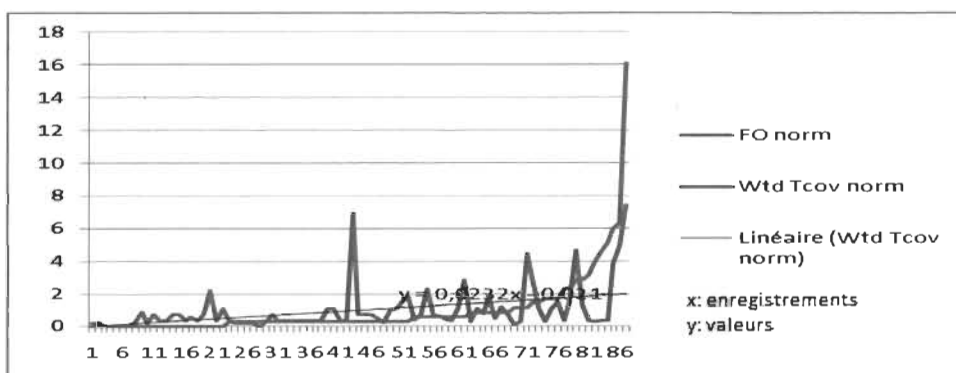


Figure 4 - Variation du taux de couverture vs la métrique FO (CAT-1)

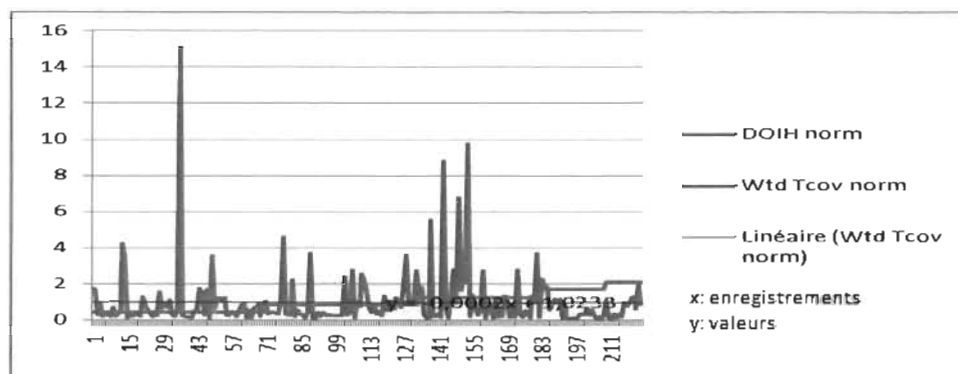


Figure 5 - Variation du taux de couverture vs la métrique DOIH (CAT-2)

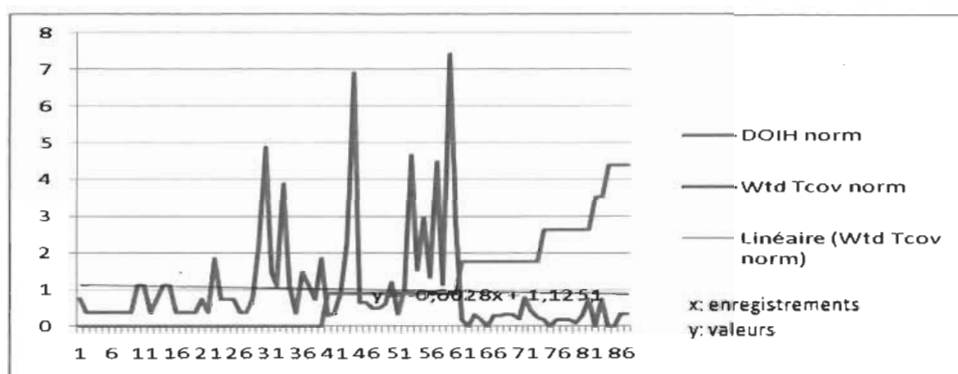


Figure 6 - Variation du taux de couverture vs la métrique DOIH (CAT-1)

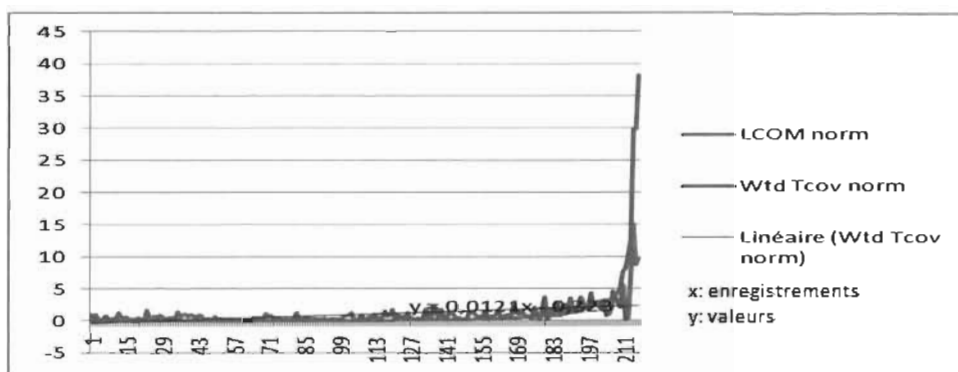


Figure 7 - Variation du taux de couverture vs la métrique LCOM (CAT-2)

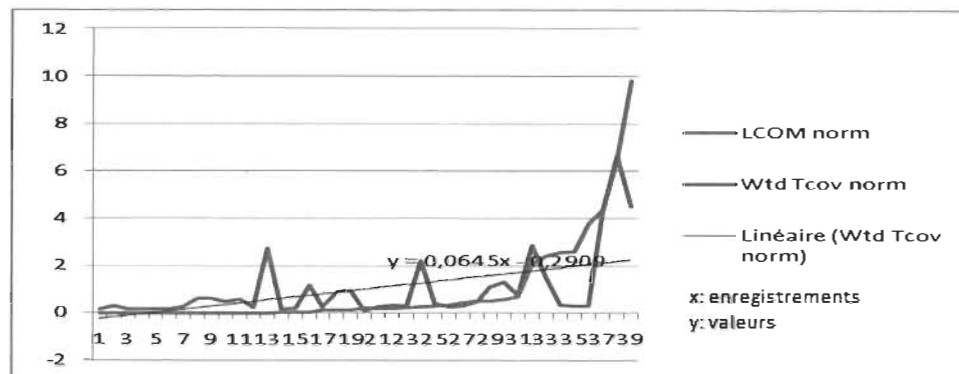


Figure 8 - Variation du taux de couverture vs la métrique LCOM (CAT-1)

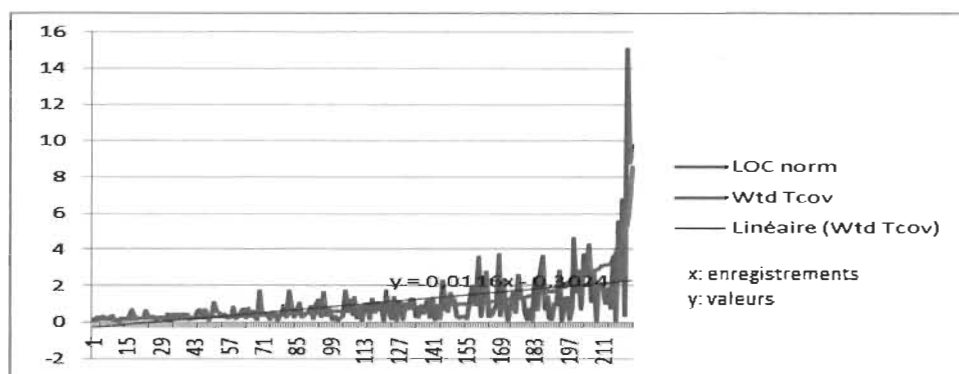


Figure 9 - Variation du taux de couverture vs la métrique LOC (CAT-2)

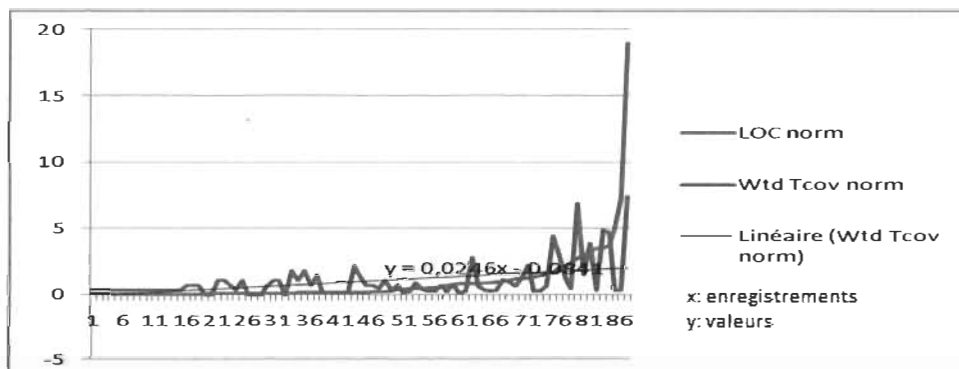


Figure 10 - Variation du taux de couverture vs la métrique LOC (CAT-1)

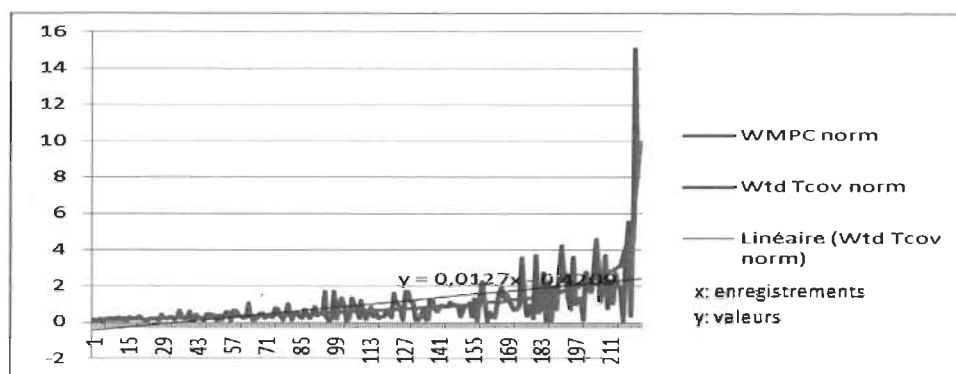


Figure 11 - Variation du taux de couverture vs la métrique WMPC (CAT-2)

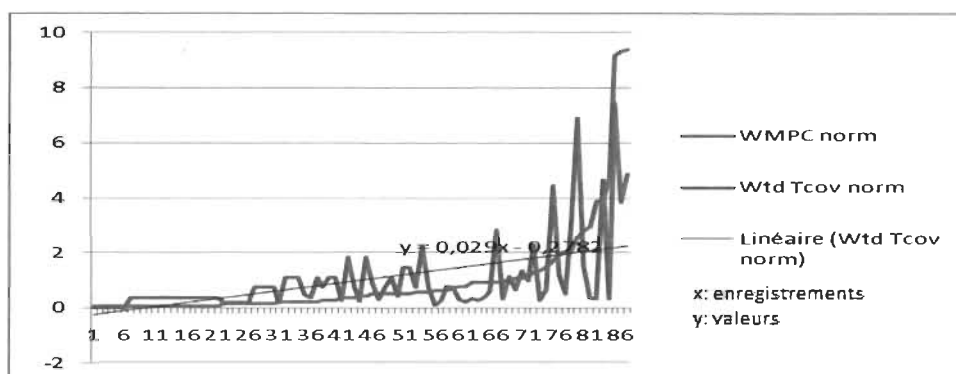


Figure 12 - Variation du taux de couverture vs la métrique WMPC (CAT-1)

Les 12 graphes de tendances présentées (*figures 1 - 12*) s'intègrent dans notre démarche expérimentale comme des éléments qui vont permettre de confirmer ou d'infirmer nos premiers constats basés sur l'analyse des statistiques descriptives et des corrélations:

- Les constats énoncés lors de la précédente étape sont bien illustrés à travers les courbes de tendances: il semblerait que les métriques COC, FO, LCOM, LOC et WMPC caractérisent les classes appartenant aux catégories CAT-2 (voir les *figures 1, 3, 7, 9 et 11*) et CAT-1 (voir les *figures 2, 4, 8, 10 et 12*) lorsque nous voyons que ces métriques

influencent positivement sur la tendance de l'étendue de la couverture des tests JUnit. Nous avons des pentes de droites de tendance positives dans tous les cas et nous remarquons également que la pente est deux fois plus forte en ce qui concerne les données des classes de CAT-1 (0,02 pour CAT-1 contre 0,01 pour CAT-2).

- Nous faisons toutefois face à une certaine contrainte au niveau des données de la métrique LCOM car au niveau des statistiques descriptives, il manque considérablement de données (48/87 pour CAT-2 contre 78/186 pour CAT-1): cette réalité nous oblige à ne pas considérer cette métrique pour ne pas biaiser nos expérimentations et nos analyses et pour conserver une certaine homogénéité au niveau de l'ensemble des données. Les valeurs manquantes sont essentiellement dues au fait que pour les classes en question, la valeur de la métrique LCOM n'est pas définie (cela est dû à la définition même de la métrique, cf. *Chapitre 2*).
- Les graphiques montrent également bien comment l'héritage à travers la métrique DOIH n'exerce aucune influence sur le taux de couverture étant donné que l'équation de la droite de tendance est quasiment une constante, comme nous pouvons le constater au niveau des *figures 5 et 6*.

Les caractéristiques qui se dégagent de nos expérimentations jusqu'ici montrent la forte et significative corrélation positive entre les métriques de couplage (FO et COC), de taille (LOC) et de complexité (WMPC) et le taux de

couverture, tandis que l'héritage (DOIH) ne semble pas affecter le taux de couverture.

4.4.1.4. Classification K-Means

À présent que nous avons identifié une série de métriques pouvant constituer des caractéristiques des catégories CAT-2 et CAT-1 du fait de leur forte corrélation avec le taux de couverture et qu'également nous avons noté que la métrique d'héritage DOIH caractérisait les classes de la catégorie CAT-0, appliquons une classification K-Means avec $k=3$ sur l'ensemble des données.

Comme décrit dans le *chapitre 2* de ce mémoire, la méthode K-Means permet de classer selon leurs caractéristiques, un ensemble de données par catégorie, et le nombre de catégories correspond à K , un nombre entier positif, comme le montre le *tableau 11*. L'outil Weka écrit en Java sera utilisé à cet effet.

Les données de métriques que nous considérerons, suite aux résultats des étapes précédentes de nos expérimentations, seront donc : FO – COC – LOC – WMPC – DOIH. Voici les résultats obtenus :

Attribute	Full Data (496)	0 (33)	1 (323)	2 (140)
COC	4.7077	19.6667	4.8483	0.8571
FO	6.8327	31.5758	3.4861	8.7214
DOIH	1.9617	2.9091	1.0217	3.9071
LOC	137.7258	769.9394	72.8885	138.2929
WMPC	27.9798	148.8788	16.7771	25.3286
Wtd_Tcov	391.4919	2216.6364	283.7678	209.8143

Tableau 11 - Synthèse de la classification k-Means

Après mise en correspondance des catégories et des clusters nous obtenons le *tableau 12* correspondant aux affectations suivantes :

	CAT-2	CAT-1	CAT-0	Total
Cluster2	66	14	60	140
Cluster1	125	72	126	323
Cluster0	32	1	0	33
Total	223	87	186	496

Tableau 12 - Correspondance des catégories et les clusters

L'analyse de ces données nous aidera à vérifier nos précédents constats:

- D'après le *tableau 12*, la classification K-Means avec $K = 3$, nous propose 3 clusters avec les caractéristiques suivantes:
 - Cluster2: comprend 140 classes (46% CAT-2 ; 1% CAT-1 et 43% CAT-0) pour lesquelles les valeurs de DOIH (métrique liée à l'héritage) sont les plus élevées (supérieur à 3,907). Par ailleurs, ce n'est pas tant la valeur presque nulle du FAN-In qui est significative, mais plus, le rapport entre le FAN-Out et le FAN-In: les classes appartenant à ce cluster font nettement plus d'appels sortants qu'elles n'ont d'appels entrants.
 - Cluster1: 323 classes (38% CAT-2, 23% CAT-1 et 39% CAT-0), qui se caractérisent essentiellement par un FAN-In plus élevé que le FAN-Out (4,83 contre 3,48) et de faibles valeurs de l'héritage (DOIH), de la complexité (WMPC) et de la taille (LOC). Nous retrouvons dans ce cluster, la majorité des classes constituant le système JFREECHART (323 / 496) dont 82,76 % des classes de CAT-1 et 67,74 % des classes de cat 0). Les classes des catégories CAT-2 et CAT-1, sont les classes pour lesquelles il n'a pas été écrit et développé des tests JUnit.

- Cluster0: 33 classes (97% CAT-2, 3% CAT-1 et 0% CAT-0) - on y trouve des valeurs élevées de presque toutes les métriques. Cela démontre bien que les classes de la catégorie CAT-2 qui forment essentiellement ce cluster sont des classes ayant une complexité particulière (WMPC = 148,87), une taille conséquente (LOC = 769, 94), un couplage conséquent dont le FAN-Out se démarque plus que le FAN-In (31,58 contre 19,67) et bien entendu, un taux de couverture élevé.
- Concernant la répartition des catégories de classes:
 - CAT-2: (30% des classes de cette catégorie se retrouvent dans le Cluster2, 56% dans le Cluster1 et 14% dans le Cluster0) - les classes de cette catégorie se retrouvent surtout dans le cluster1, qui est un reflet général des caractéristiques du système. Ceci étant, comme nous l'avons vu précédemment, cette catégorie forme à elle seule le cluster0, marqué par sa complexité.
 - CAT-1: (16% cluster2, 83% cluster1 et 1% cluster0) - les classes de la catégorie CAT-1 ne se caractérisent pas par un FAN-In faible (élément représentatif du cluster0 où elles sont quasi absentes). Bien au contraire, la grande partie des classes de cette catégorie se situe dans le cluster1, dont le FAN-In est plus important que le FAN-Out, ce qui vient étayer l'explication selon laquelle ces classes sont couvertes par les tests JUnit, car possédant des appels entrants conséquents.
 - CAT-0: (33% cluster2, 67% cluster1 et 0% cluster0) - on pourrait penser que ces classes présentent les mêmes caractéristiques que les classes de CAT-1: néanmoins, le tiers

de ces classes se trouvent dans le cluster2, qui se démarque par son héritage élevé et son FAN-In, presque nul.

L'analyse par classification K-Means fournit des résultats en total accord avec les expérimentations des étapes antérieures, comme le montre le *tableau 9*, à savoir:

- la complexité, la taille et le couplage élevés sont des caractéristiques des classes de CAT-2,
- les classes de la catégorie CAT-1 dénotent un FAN-In plus important que le FAN-Out,
- les classes de la catégorie CAT-0, outre des valeurs plus faibles quant aux métriques de complexité et de taille, possèdent des valeurs de métriques DOIH (héritage) non négligeables.

4.4.2 JODA

4.4.2.1. Analyse des statistiques descriptives

Tout comme nous l'avons fait avec le système JFREECHART, nous allons en premier lieu, procéder à une analyse des statistiques descriptives des données de métriques de JODA, pour chacune des catégories CAT-2, CAT-1 et CAT-0. Cela va permettre de dégager quelques éléments caractéristiques de chaque système.

Les *tableaux 13, 14 et 15* correspondants aux statistiques descriptives par catégorie de classes sont présentés ci-après:

Variable	Observation	Minimum	Maximum	Moyenne
CBO	75	1,000	29,000	10,800
COC	75	0,000	57,000	7,027
FO	75	1,000	22,000	7,547
DOIH	75	1,000	6,000	2,413
NOCC	75	0,000	25,000	1,173
LCOM	70	0,000	6582,000	743,686
LOC	75	14,000	1760,000	234,000
NOA	75	0,000	59,000	5,800
NOO	75	2,000	103,000	25,773
RFC	75	17,000	287,000	107,813
WMPC	75	3,000	176,000	45,640
WTcov	75	66,000	9270,000	2229,040

Tableau 13 - Statistiques descriptives des données des classes de CAT-2

Variable	Observation	Minimum	Maximum	Moyenne
CBO	96	0,000	36,000	6,802
COC	96	0,000	48,000	2,865
FO	96	0,000	22,000	5,052
DOIH	96	0,000	4,000	1,333
NOCC	96	0,000	32,000	2,594
LCOM	67	0,000	1225,000	81,015
LOC	96	3,000	1161,000	109,375
NOA	96	0,000	39,000	2,604
NOO	96	0,000	50,000	11,500
RFC	96	1,000	260,000	54,865
WMPC	96	1,000	89,000	22,156
WTcov	96	0,000	5000,000	614,521

Tableau 14 - Statistiques descriptives des données des classes de CAT-1

Variable	Observation	Minimum	Maximum	Moyenne
CBO	54	0,000	10,000	3,704
COC	54	0,000	27,000	1,315
FO	54	0,000	5,000	1,944
DOIH	54	0,000	5,000	2,556

Variable	Observation	Minimum	Maximum	Moyenne
NOCC	54	0,000	7,000	0,926
LCOM	44	0,000	325,000	45,909
LOC	54	3,000	247,000	65,574
NOA	54	0,000	40,000	3,463
NOO	54	0,000	51,000	10,519
RFC	54	2,000	212,000	88,796
WMPC	54	1,000	69,000	16,963
WTcov	54	0,000	0,000	0,000

Tableau 15 - Statistiques descriptives des données des classes de CAT-0

L'analyse des *tableaux 13, 14 et 15* fait ressortir un certain nombre de points à savoir:

- JODA compte 225 classes dont le tiers (33%) appartiennent à la catégorie CAT-2, près de la moitié (43%) appartiennent à la catégorie CAT-1 et 24% à CAT-0: déjà, avec la proportion des classes de CAT-0, nous pouvons noter qu'une partie non négligeable de ce système n'a pas été testée (67%) (pas de tests JUnit développés) et le quart n'est pas atteint (couvert) par les tests des autres classes;
- Les classes de la catégorie CAT-2 présentent des valeurs relativement élevées pour ce qui est de la complexité (WMPC et RFC: 45 et 107 en moyenne), la taille (LOC, NOA et NOO), le couplage sortant (CBO, FO) et la cohésion, comme nous le montre le *tableau 13*. Rappelons qu'il s'agit ici des classes pour lesquelles des tests JUnit ont été développés (explicitement) et dont bien entendu nous disposons de valeurs du taux de couverture pondéré non nulles voire même relativement élevées (moyenne: 2229,04). En partant de la formule de calcul du taux de couverture pondéré (cf. *Chapitre 3*) et en notant que

le nombre d'opérations (métrique NOO) en moyenne pour chaque classe de la CAT-2 est de 25,77, on peut avancer qu'en moyenne 86,50% ($2229,04 / 25,77$) des méthodes de chaque classe sont appelées lors des tests, ce qui est vraiment intéressant car cela représente près de la totalité des méthodes d'une classe qui est appelée par d'autres classes lors des tests: cela nous amène à porter un intérêt particulier ici aussi aux métriques de couplage;

- Au niveau du *tableau 14*, ce qui serait le plus notable quant aux classes de la catégorie CAT-1, ce serait le fait que la valeur maximale du couplage entrant (FAN-In ou COC) est de 48 contre 22 pour le couplage sortant (Fan-Out); néanmoins, nous constatons une moyenne des valeurs plus petite en ce qui concerne le FAN-In et le FAN-Out (2,86 versus 5,05).
- Concernant les classes de la catégorie CAT-0 (*tableau 15*), force est de constater qu'elles présentent un couplage (CBO, FO, COC) relativement faible (entre 1,31 et 3,70), en plus de valeurs des métriques de taille (LOC, NOA, NOO) toutes aussi relativement faibles comparativement aux valeurs de ces 3 métriques de taille pour les classes des catégories CAT-2 et CAT-1. Ces classes n'ont pas été testées et ne sont pas atteintes par les tests d'autres classes bien qu'elles représentent 24% des classes que compte le système.

Il est très notable que les classes des catégories CAT-2 et CAT-0 s'opposent du point de vue de leurs caractéristiques. Tandis que celles de CAT-2 ont des valeurs élevées des métriques de taille, de complexité et de couplage (nous

laisserons de côté les métriques de cohésion du fait du manque de données non négligeable), celles de CAT-0 ont des valeurs très faibles des données des mêmes métriques.

Ainsi, les familles de métriques à suivre de près seraient la taille, la complexité et le couplage: les métriques de cohésion étant biaisées par le fait qu'il nous manque considérablement des données et les métriques d'héritage ne semblant pas être une caractéristique particulière distinguant les classes entre elles.

4.4.2.2. Analyse des corrélations (Spearman et Pearson)

L'analyse de corrélations de Spearman et de Pearson entre les métriques orientées-objet et le taux de couverture, pour chacune des catégories CAT-2 et CAT-1 servira à déterminer la significativité ou non de la corrélation entre une métrique donnée et le taux de couverture afin de faire ressortir une ou des caractéristiques du système. Notons que nous ne considérerons pas, lors de cette étape, les classes de la catégorie CAT-0 car elles n'ont pas de valeurs définies du taux de couverture WT_{cov} .

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WT_{cov}
----------	-----	-----	----	------	------	------	-----	-----	-----	-----	------	------------

CBO	1	0,205	0,927	0,159	-0,216	0,708	0,658	0,460	0,645	0,584	0,678	0,632
COC	0,205	1	0,243	-0,367	0,359	0,394	0,400	0,264	0,450	0,111	0,394	0,439
FO	0,927	0,243	1	0,067	-0,168	0,753	0,728	0,391	0,716	0,568	0,750	0,690
DOIH	0,159	-0,367	0,067	1	-0,275	0,102	-0,020	0,114	0,052	0,663	0,001	-0,044
NOCC	-0,216	0,359	-0,168	-0,275	1	-0,023	-0,074	-0,158	0,050	-0,112	-0,015	0,029
LCOM	0,708	0,394	0,753	0,102	-0,023	1	0,840	0,435	0,948	0,670	0,887	0,862
LOC	0,658	0,400	0,728	-0,020	-0,074	0,840	1	0,571	0,860	0,598	0,932	0,807
NOA	0,460	0,264	0,391	0,114	-0,158	0,435	0,571	1	0,489	0,502	0,454	0,411
NOO	0,645	0,450	0,716	0,052	0,050	0,948	0,860	0,489	1	0,634	0,936	0,923
RFC	0,584	0,111	0,568	0,663	-0,112	0,670	0,598	0,502	0,634	1	0,596	0,485
WMPC	0,678	0,394	0,750	0,001	-0,015	0,887	0,932	0,454	0,936	0,596	1	0,889
WTcov	0,632	0,439	0,690	-0,044	0,029	0,862	0,807	0,411	0,923	0,485	0,889	1

Tableau 16 - Corrélations de Spearman des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,069	0,944	0,272	-0,113	0,435	0,734	0,401	0,534	0,638	0,679	0,371
COC	0,069	1	0,074	-0,065	0,664	0,401	0,063	-0,205	0,540	0,209	0,259	0,578
FO	0,944	0,074	1	0,168	-0,129	0,325	0,633	0,319	0,551	0,543	0,615	0,389
DOIH	0,272	-0,065	0,168	1	-0,282	0,185	0,460	0,504	0,227	0,739	0,268	0,147
NOCC	-0,113	0,664	-0,129	-0,282	1	0,375	-0,259	-0,473	0,249	-0,106	-0,015	0,285
LCOM	0,435	0,401	0,325	0,185	0,375	1	0,643	0,197	0,499	0,518	0,629	0,118
LOC	0,734	0,063	0,633	0,460	-0,259	0,643	1	0,594	0,541	0,776	0,888	0,332
NOA	0,401	-0,205	0,319	0,504	-0,473	0,197	0,594	1	0,119	0,516	0,409	0,037
NOO	0,534	0,540	0,551	0,227	0,249	0,499	0,541	0,119	1	0,674	0,760	0,720
RFC	0,638	0,209	0,543	0,739	-0,106	0,518	0,776	0,516	0,674	1	0,736	0,477
WMPC	0,679	0,259	0,615	0,268	-0,015	0,629	0,888	0,409	0,760	0,736	1	0,460
WTcov	0,371	0,578	0,389	0,147	0,285	0,118	0,332	0,037	0,720	0,477	0,460	1

Tableau 17 - Corrélations de Spearman des métriques OO des classes de CAT-1

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	-0,007	0,890	0,117	-0,166	0,510	0,506	0,167	0,596	0,515	0,620	0,606
COC	-0,007	1	-0,012	-0,397	0,310	-0,017	0,031	0,127	0,109	-0,198	0,094	0,135
FO	0,890	-0,012	1	0,009	-0,139	0,634	0,655	0,132	0,718	0,514	0,699	0,721
DOIH	0,117	-0,397	0,009	1	-0,200	0,120	-0,104	-0,047	0,048	0,727	-0,009	-0,016
NOCC	-0,166	0,310	-0,139	-0,200	1	0,004	-0,071	-0,150	0,107	-0,101	0,059	0,122
LCOM	0,510	-0,017	0,634	0,120	0,004	1	0,602	0,147	0,926	0,582	0,852	0,904
LOC	0,506	0,031	0,655	-0,104	-0,071	0,602	1	0,423	0,728	0,506	0,769	0,684
NOA	0,167	0,127	0,132	-0,047	-0,150	0,147	0,423	1	0,327	0,216	0,435	0,313
NOO	0,596	0,109	0,718	0,048	0,107	0,926	0,728	0,327	1	0,602	0,939	0,956
RFC	0,515	-0,198	0,514	0,727	-0,101	0,582	0,506	0,216	0,602	1	0,550	0,504
WMPC	0,620	0,094	0,699	-0,009	0,059	0,852	0,769	0,435	0,939	0,550	1	0,926
WTcov	0,606	0,135	0,721	-0,016	0,122	0,904	0,684	0,313	0,956	0,504	0,926	1

Tableau 18 - Corrélations de Pearson des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	-0,037	0,884	0,242	-0,100	0,317	0,673	0,290	0,390	0,606	0,664	0,181
COC	-0,037	1	-0,002	-0,074	0,761	0,627	0,020	0,135	0,537	0,109	0,230	0,585
FO	0,884	-0,002	1	0,110	-0,128	0,081	0,675	0,151	0,310	0,467	0,538	0,248
DOIH	0,242	-0,074	0,110	1	-0,173	0,342	0,259	0,357	0,239	0,753	0,232	0,107
NOCC	-0,100	0,761	-0,128	-0,173	1	0,830	-0,074	0,077	0,489	0,034	0,150	0,469
LCOM	0,317	0,627	0,081	0,342	0,830	1	0,318	0,572	0,882	0,579	0,567	0,299
LOC	0,673	0,020	0,675	0,259	-0,074	0,318	1	0,353	0,325	0,644	0,717	0,178
NOA	0,290	0,135	0,151	0,357	0,077	0,572	0,353	1	0,400	0,422	0,539	0,232
NOO	0,390	0,537	0,310	0,239	0,489	0,882	0,325	0,400	1	0,568	0,653	0,680
RFC	0,606	0,109	0,467	0,753	0,034	0,579	0,644	0,422	0,568	1	0,589	0,257
WMPC	0,664	0,230	0,538	0,232	0,150	0,567	0,717	0,539	0,653	0,589	1	0,384
WTcov	0,181	0,585	0,248	0,107	0,469	0,299	0,178	0,232	0,680	0,257	0,384	1

Tableau 19 - Corrélations de Pearson des métriques OO des classes de CAT-1

Après avoir généré les données de corrélation de Spearman et de Pearson des métriques orientées-objet et du taux de couverture à l'aide de l'outil XLSTAT, passons à présent à l'analyse respective des *tableaux 16, 17, 18 et 19* :

- Les *tableaux 16 et 17* révèlent bien à quel point la corrélation entre le FAN-In et le taux de couverture est significative pour la catégorie CAT-1 (0,57 versus 0,43 pour CAT-2: corrélation de Spearman).
- Dans les deux catégories, nous voyons que l'héritage (à travers DOIH) n'est pas corrélé au taux de couverture.
- La cohésion (LCOM) également est fortement corrélée au taux de couverture avec une plus forte intensité en faveur de CAT-2, tel que nous pouvons le constater dans les *tableaux 16 et 18*.
- Le *tableau 18* nous indique que la complexité quant à elle, via la métrique WMPC, est significativement corrélée au taux de couverture dans la catégorie CAT-2 (CAT-2: 0,92 pour Pearson).
- Il est tout à fait normal que les corrélations entre la métrique NOO et le taux de couverture soient fortes ($> 0,68$) car le calcul du taux de couverture s'est fait à partir des valeurs de NOO.
- Notons également qu'au niveau de la catégorie CAT-1, la métrique NOA n'a pas une corrélation significative avec le taux de couverture (*tableaux 16 et 19*), et cela peut se constater dans les statistiques descriptives (*tableau 14*), où la moyenne des valeurs de NOA est relativement plus faible dans la catégorie CAT-1 comparée à celle trouvée dans la catégorie CAT-2 dans le *tableau 13* (2,60 contre 5,80). Ce facteur pourrait être déterminant quant au fait qu'une classe soit testée ou pas. Si nous nous penchons sur la métrique LOC, nous voyons

bien que cette métrique est significativement corrélée au taux de couverture en ce qui concerne la catégorie CAT-2, tel que nous pouvons le constater dans les *tableaux 19 et 17* (0,68 et 0,80 respectivement Pearson et Spearman) contrairement au cas de la catégorie CAT-1 (0,18 et 0,33 respectivement Pearson et Spearman) dans les *tableaux 18 et 16*: la métrique LOC pourrait donc aussi être déterminante dans la décision de tester une classe ou pas.

L'analyse des corrélations révèle un certain nombre d'informations supplémentaires quant aux caractéristiques des classes pour lesquelles des tests JUnit ont été développés, mais également pour celles pour lesquelles nous avons des données de taux de couverture non nulles alors qu'il n'y a pas eu de tests développées pour ces classes: en effet, les classes de la catégorie CAT-1 possèdent un FAN-In fortement corrélé au taux de couverture, tandis que les classes de CAT-2 se démarquent surtout par les métriques LOC et NOA, plus corrélées au taux de couverture que celles de CAT-1. Ces caractéristiques de taille pourraient être des facteurs dans la détermination des tests à développer.

Une caractéristique commune aux deux catégories, serait la forte corrélation positive de la métrique de complexité WMPC ainsi que de la métrique de cohésion LCOM avec le taux de couverture.

4.4.2.3. Analyse des courbes de tendances

En sélectionnant quelques métriques (COC, FO, DOIH, LOC et WMPC) à la suite de l'étape expérimentale précédente, tâchons d'analyser les tendances de

variation des valeurs des métriques et du taux de couverture dans chaque catégorie.

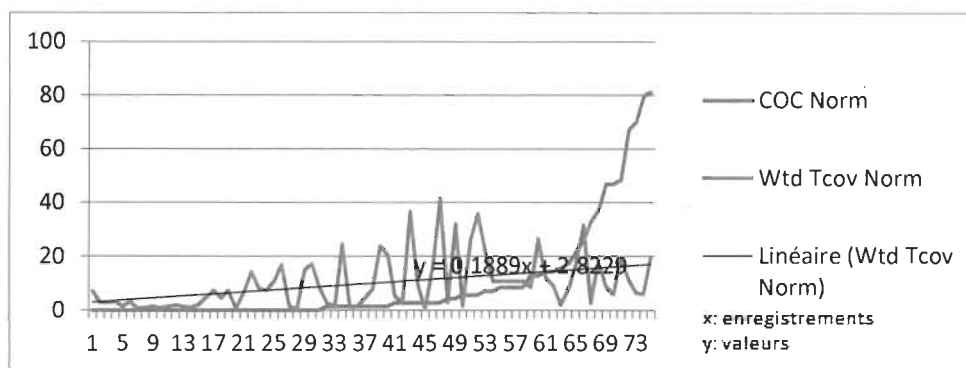


Figure 13 - Variation du taux de couverture vs la métrique COC (CAT-2)

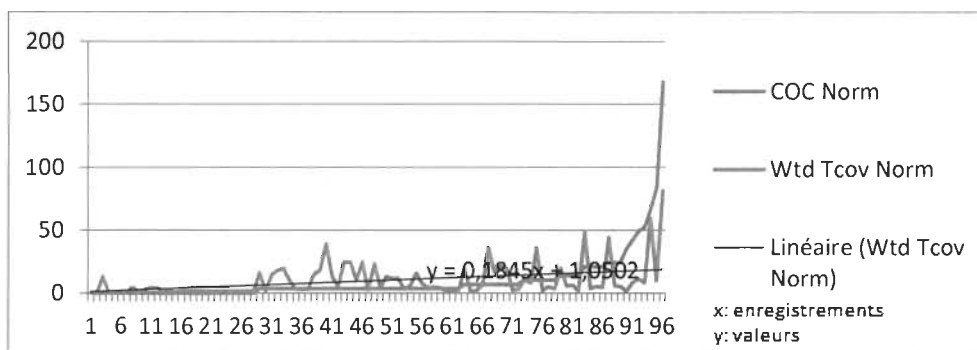


Figure 14 - Variation du taux de couverture vs la métrique COC (CAT-1)

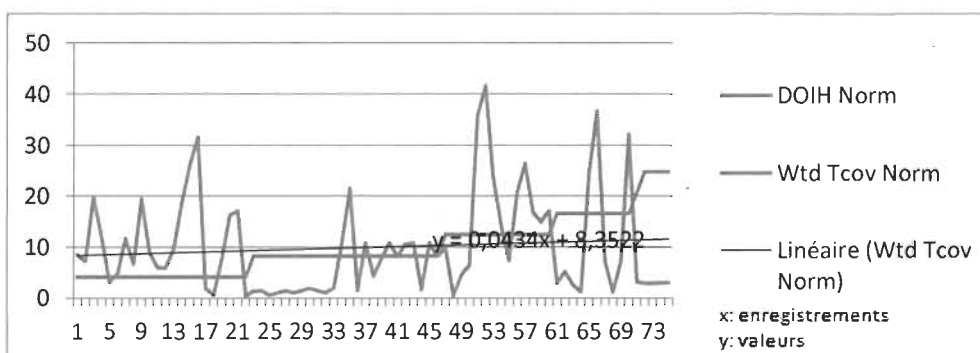


Figure 15 - Variation du taux de couverture vs la métrique FO (CAT-2)

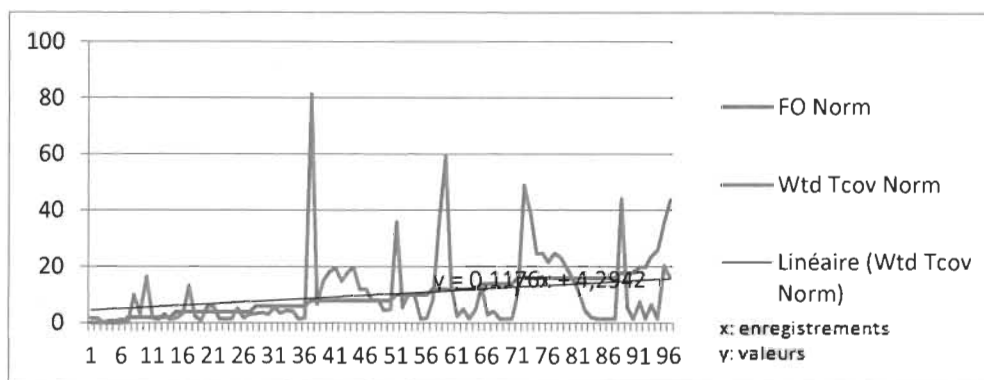


Figure 16 - Variation du taux de couverture vs la métrique FO (CAT-1)

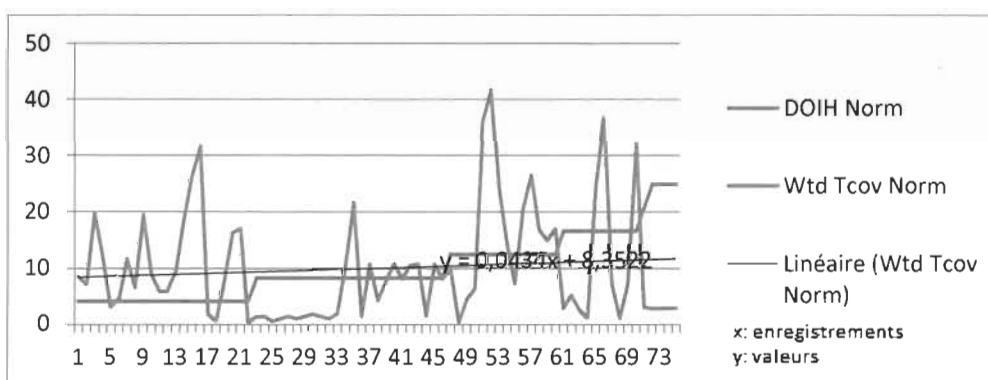


Figure 17 - Variation du taux de couverture vs la métrique DOIH (CAT-2)

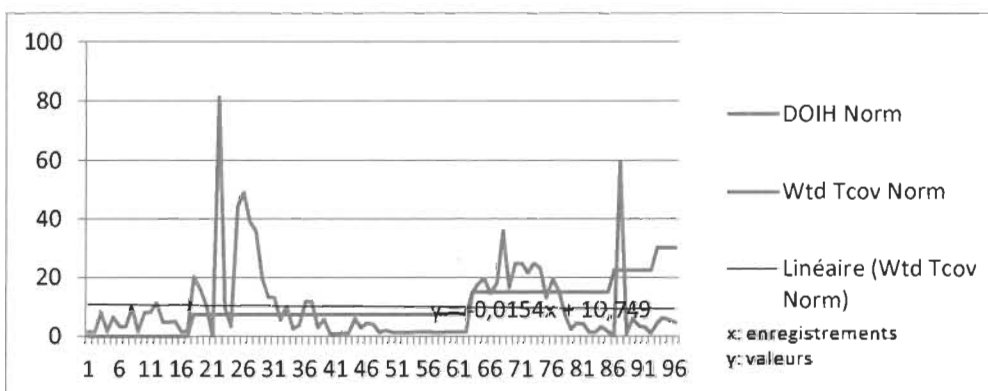


Figure 18 - Variation du taux de couverture vs la métrique DOIH (CAT-1)

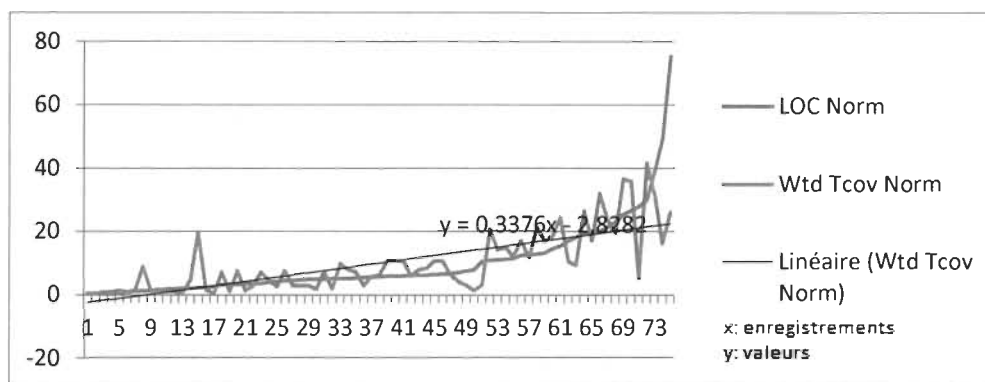


Figure 19 - Variation du taux de couverture vs la métrique LOC (CAT-2)

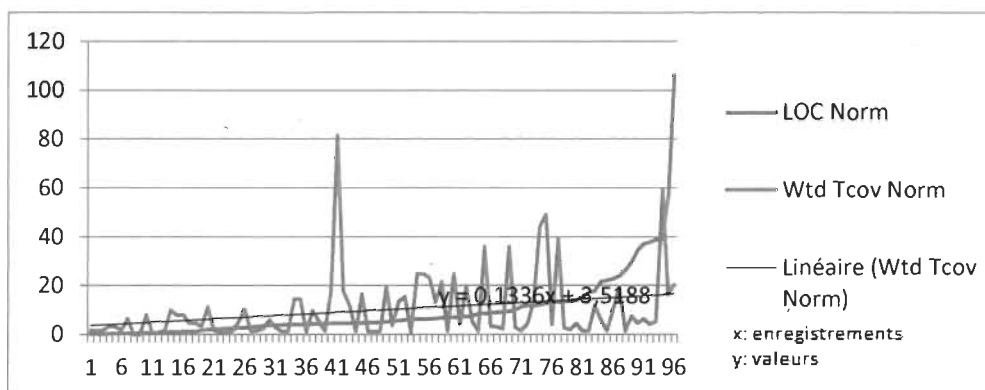


Figure 20 - Variation du taux de couverture vs la métrique LOC (CAT-1)

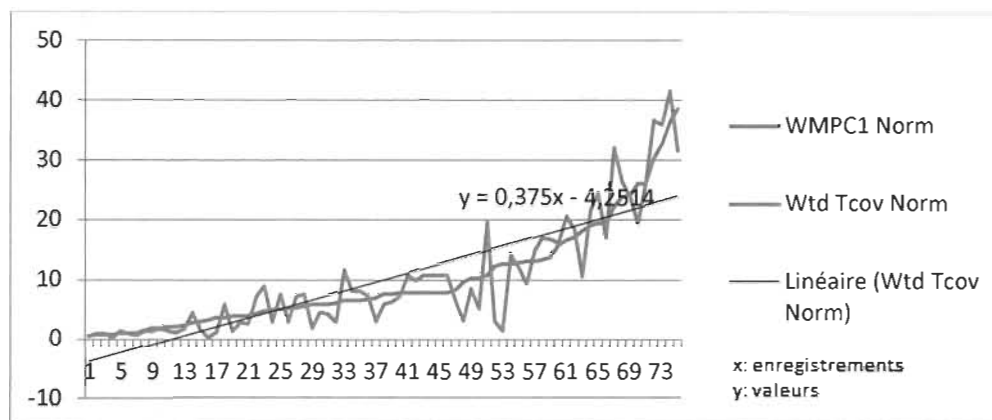


Figure 21 - Variation du taux de couverture vs la métrique WMPC (CAT-2)

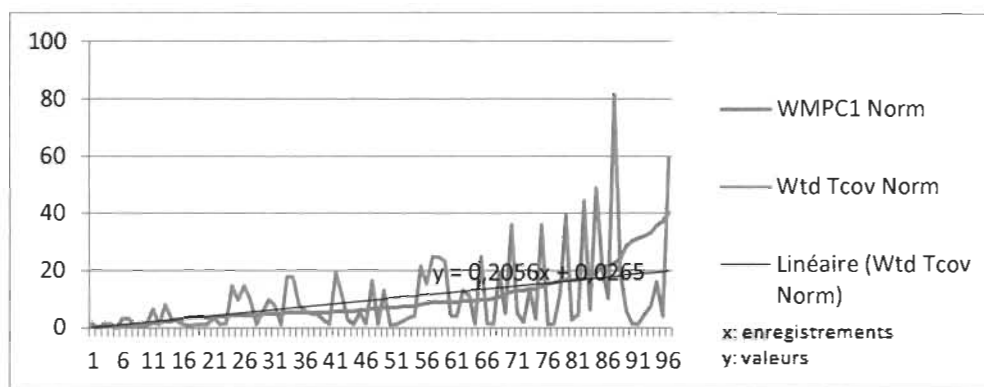


Figure 22 - Variation du taux de couverture vs la métrique WMPC (CAT-1)

Les figures 13 à 22 s'intègrent dans notre démarche expérimentale comme des éléments qui vont permettre de confirmer ou d'infirmer nos premiers constats basés sur l'analyse des statistiques descriptives et des corrélations:

- Une validation partielle des constats énoncés lors de la précédente étape est faite à travers les courbes de tendances: en effet, il semblerait que les métriques COC, LOC et WMPC caractérisent les classes appartenant aux catégories CAT-2 et CAT-1 lorsque nous voyons que ces métriques influencent positivement sur la tendance de l'étendue de

la couverture des tests JUnit comme nous le montrent les *figures 13, 14, 19, 20, 21 et 22*. Nous avons des pentes de droites de tendance positives et relativement conséquentes, de l'évolution du taux de couverture en fonction de l'évolution des valeurs de ces dernières métriques: si nous ignorons la pente de la droite de la tendance du taux de couverture par rapport aux valeurs de la métrique LOC pour les classes de la catégorie CAT-1 (qui est de 0,1336, *figure 20*), les pentes varient de 0,1854 (pour COC versus le taux de couverture pour les classes de CAT-1, *figure 14*) à 0,375 (pour WMPC versus le taux de couverture pour les classes de CAT-2, *figure 22*).

- Les pentes des droites de tendance de l'évolution du taux de couverture par rapport aux données de la métrique FO sont relativement faibles comparées aux pentes citées précédemment (pour CAT-2 dans la *figure 15*, nous avons 0,0434 et pour CAT-1 0,1176, *figure 16*); remarquons tout de même que la pente de la droite relative aux classes de CAT-1 est plus de 2 fois supérieure à celle de la droite relative aux classes de CAT-2.
- Les graphiques montrent également bien comment l'héritage à travers la métrique DOIH n'exerce aucune influence sur le taux de couverture étant donné que l'équation de la droite de tendance est quasiment une constante, comme nous pouvons le constater à travers les *figures 17 et 18*.

Les caractéristiques qui se dégagent de nos expérimentations jusqu'ici montrent la forte et significative corrélation positive entre les métriques de

couplage (FO et COC), de taille (LOC) et de complexité (WMPC) et le taux de couverture, tandis que l'héritage (DOIH) ne semble pas affecter le taux de couverture.

4.4.2.4. Classification K-Means

À présent que nous avons identifié une série de métriques pouvant constituer des caractéristiques des catégories CAT-2 et CAT-1 du fait de leur forte corrélation avec le taux de couverture et qu'également nous avons noté que la métrique d'héritage DOIH caractérisait les classes de la catégorie CAT-0, appliquons une classification K-Means avec $k=3$ sur l'ensemble des données.

Les données de métriques que nous considérerons, suite aux résultats des étapes précédentes de nos expérimentations, seront: FO – COC – LOC – WMPC – DOIH.

Voici les résultats obtenus:

Attribute	Full Data (225)	0 (33)	1 (91)	2 (101)
COC	3.88	6.9394	5.8352	1.1188
FO	5.1378	13.7576	3.8901	3.4455
DOIH	1.9867	2.303	0.7363	3.0099
LOC	140.4044	478.7879	75.8132	88.0396
WMPC	28.7378	80.2727	19.1868	20.505
Wtd_Tcov	1266.7867	5347.6667	683.6813	458.802

Tableau 20 - Synthèse de la classification k-means

Après mise en correspondance des catégories et des clusters nous obtenons le tableau d'affectation suivant:

	CAT-2	CAT-1	CAT-0	Total
Cluster2	33	31	37	101
Cluster1	15	59	17	91
Cluster0	27	6	0	33
Total	75	96	54	225

Tableau 21 - Correspondance des catégories et les clusters

L'analyse de ces données nous aidera à vérifier nos précédents constats:

- D'après le *tableau 21*, la classification K-Means avec $K = 3$, nous propose 3 clusters avec les caractéristiques suivantes:
 - Cluster2: comprend 101 classes (37% CAT-2 ; 30% CAT-1 et 33% CAT-0) pour lesquelles les valeurs de DOIH (métrique liée à l'héritage) sont les plus élevées (supérieur à 3,009). Par ailleurs, ce n'est pas tant la faible valeur du FAN-In qui est significative (1,119), mais plus le rapport entre le FAN-Out et le FAN-In : les classes appartenant à ce cluster font nettement plus d'appels sortants qu'elles n'ont d'appels entrants (environ 3 fois plus).
 - Cluster1 : 91 classes (16% CAT-2, 65% CAT-1 et 19% CAT-0), qui se caractérisent essentiellement par un FAN-In plus élevé que le FAN-Out (5,83 contre 3,89) et les plus faibles valeurs de l'héritage (DOIH), de la complexité (WMPC) et de la taille (LOC). Plus de 60% des classes retrouvées dans ce cluster sont issues de la catégorie CAT-1.
 - Cluster0 : 33 classes (82% CAT-2, 18% CAT-1 et 0% CAT-0)
 - on y trouve des valeurs élevées de presque toutes les métriques. Cela démontre bien que les classes de la catégorie CAT-2 qui forment essentiellement ce cluster sont des classes

ayant une complexité particulière (WMPC = 80,27), une taille conséquente (LOC = 478,79), un couplage conséquent dont le FAN-Out se démarque plus que le FAN-In (13,76 contre 6,94) et bien entendu, un taux de couverture élevé.

- Concernant la répartition des catégories de classes:
 - CAT-2: (44% des classes de cette catégorie se retrouvent dans le Cluster2, 20% dans le Cluster1 et 36% dans le Cluster0).
 - CAT-1: (33% cluster2, 61% cluster1 et 6% cluster0) - plus de 60% des classes de cette catégorie sont retrouvées dans le cluster1, cluster caractérisé par des valeurs de FAN-In nettement supérieures aux valeurs de FAN-Out. Ce constat vient étayer l'explication selon laquelle ces classes (de la catégorie CAT-1) sont couvertes par les tests JUnit, car elles possèdent des appels entrants conséquents.
 - CAT-0: (69% cluster2, 31% cluster1 et 0% cluster0) - le fait qu'aucune classe de la catégorie CAT-0 n'appartienne au cluster0 où l'on retrouve des valeurs élevées de toutes les métriques sélectionnées (COC, FO, LOC, WMPC, WTcov) à l'exception de DOIH, conforte l'hypothèse selon laquelle les classes de la catégorie CAT-0 sont des classes ayant de faibles valeurs de métriques à l'exception des métriques d'héritage (en l'occurrence DOIH).

L'analyse par classification K-Means fournit des résultats en total accord avec les expérimentations des étapes antérieures, à savoir:

- la complexité, la taille et le couplage élevés sont des caractéristiques des classes de CAT-2,
- les classes de la catégorie CAT-1 dénotent un FAN-In plus important que le FAN-Out,
- les classes de la catégorie CAT-0, outre des valeurs plus faibles quant aux métriques de complexité et de taille, possèdent des valeurs de métriques DOIH (héritage) non négligeables.

4.4.3 IO

4.4.3.1. Analyse des statistiques descriptives

Tel que précédemment, procédons tout d'abord à une analyse des statistiques descriptives des données de métriques de IO, pour chacune des catégories CAT-2, CAT-1 et CAT-0. Cela va permettre de dégager quelques éléments caractéristiques de chaque système.

Les *tableaux 22, 23 et 24* aux statistiques descriptives par catégorie de classes sont représentés ci-après:

Variable	Observation	Minimum	Maximum	Moyenne
CBO	59	0,000	39,000	4,695
COC	59	0,000	10,000	0,780
FO	59	0,000	21,000	3,000
DOIH	59	1,000	5,000	2,305
NOCC	59	0,000	4,000	0,102
LCOM	49	0,000	3916,000	163,224
LOC	59	7,000	968,000	102,593
NOA	59	0,000	19,000	3,441
NOO	59	0,000	88,000	10,102
RFC	59	20,000	202,000	44,119
WMPC	59	1,000	250,000	24,186
WTcov	59	0,000	8184,000	836,373

Tableau 22 - Statistiques descriptives des données des classes de CAT-2

Variable	Observation	Minimum	Maximum	Moyenne
CBO	25	1,000	26,000	3,600
COC	25	0,000	6,000	0,880
FO	25	1,000	8,000	2,120
DOIH	25	0,000	4,000	1,840
NOCC	25	0,000	22,000	1,680
LCOM	19	0,000	903,000	59,632
LOC	25	4,000	194,000	47,040
NOA	25	0,000	8,000	1,360
NOO	25	0,000	42,000	5,680
RFC	25	4,000	89,000	28,600
WMPC	25	2,000	60,000	12,600
WTcov	25	0,000	3696,000	408,040

Tableau 23 - Statistiques descriptives des données des classes de CAT-1

Variable	Observation	Minimum	Maximum	Moyenne
CBO	25	1,000	26,000	4,440
COC	25	0,000	11,000	0,760
FO	25	1,000	11,000	2,600
DOIH	25	0,000	4,000	1,800
NOCC	25	0,000	22,000	2,520
LCOM	18	0,000	903,000	77,278
LOC	25	7,000	1642,000	193,680
NOA	25	0,000	8,000	1,840
NOO	25	1,000	42,000	5,800
RFC	25	0,000	45,000	17,440
WMPC	25	1,000	62,000	12,600

Tableau 24 - Statistiques descriptives des données des classes de CAT-0

L'analyse des *tableaux 22, 23, 24* fait ressortir un certain nombre de points à savoir:

- IO compte 109 classes dont plus de la moitié (54%) appartiennent à la catégorie CAT-2 (*tableau 22*), tandis que les catégories CAT-1 et CAT-0 représentent chacun 23%, soit le quart, des classes que compte ce système (*tableaux 23, 24*): ainsi, une partie considérable du système IO a été testée ou est concernée par les tests JUnit qui ont été développés (environ 77%);
- Les classes de la catégorie CAT-2 présentent les valeurs les plus élevées pour ce qui est de la complexité (WMPC et RFC: 24 et 44 en moyenne), la taille (LOC, NOA et NOO), le couplage sortant (CBO, FO) et la cohésion, tel que le montre le *tableau 22*. Rappelons qu'il s'agit ici des classes pour lesquelles des tests JUnit ont été développés et dont bien entendu nous disposons de valeurs du taux de couverture pondéré non nulles voire même relativement élevées (moyenne: 836)

comparativement aux valeurs moyennes trouvées au niveau de la catégorie CAT-1, (408, *tableau 23*). En partant de la formule de calcul du taux de couverture pondéré (cf. *Chapitre 3*) et en notant que le nombre d'opérations (métrique NOO) en moyenne pour chaque classe de la CAT-2 est de 10,10 on peut avancer qu'en moyenne 82,80% ($836,37 / 10,10$) des méthodes de chaque classe est appelée lors des tests, ce qui est vraiment intéressant car cela représente près de la totalité des méthodes d'une classe qui est appelée par d'autres classes lors des tests: cela nous amène à porter un intérêt particulier aux métriques de couplage;

- Concernant les classes de la catégorie CAT-1 et CAT-0 (*tableau 23 et 24*), celles-ci présentent plus ou moins les mêmes caractéristiques si l'on se base sur les données des métriques au niveau des statistiques descriptives; néanmoins, les classes de la catégorie CAT-0 présentent une moyenne de lignes de code notablement plus élevée (193 pour CAT-0 contre 47 pour CAT-1).

L'analyse des statistiques descriptives du système fait clairement ressortir les classes de la catégorie CAT-2 qui diffèrent des classes des catégories CAT-1 et CAT-0 du point de vue de leurs caractéristiques. Tandis que celles de CAT-2 ont des valeurs élevées des métriques de taille, de complexité et de couplage (nous laisserons de côté les métriques de cohésion du fait du manque de données non négligeable), celles de CAT-1 et CAT-0 ont des valeurs très faibles des données des mêmes métriques.

Ainsi, les familles de métriques à suivre de près seraient la taille, la complexité et le couplage; les métriques de cohésion étant biaisées par le fait qu'il nous manque considérablement des données et les métriques d'héritage ne semblant pas être une caractéristique particulière distinguant les classes entre elles.

4.4.3.2. Analyse des corrélations (Spearman et Pearson)

L'analyse de corrélations de Spearman et de Pearson entre les métriques orientées-objet et le taux de couverture, pour chacune des catégories CAT-2 et CAT-1 servira à déterminer la significativité ou non de la corrélation entre une métrique donnée et le taux de couverture afin de faire ressortir une ou des caractéristiques du système.

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,055	0,963	-0,396	-0,169	0,579	0,628	0,412	0,421	0,553	0,599	0,353
COC	0,055	1	0,053	-0,291	0,153	0,285	0,326	0,382	0,383	0,124	0,355	0,488
FO	0,963	0,053	1	-0,426	-0,154	0,628	0,681	0,445	0,468	0,531	0,630	0,395
DOIH	-0,396	-0,291	-0,426	1	0,066	-0,386	-0,473	-0,423	-0,384	0,079	-0,514	-0,372
NOCC	-0,169	0,153	-0,154	0,066	1	-0,056	-0,150	-0,154	-0,153	-0,276	-0,191	-0,077
LCOM	0,579	0,285	0,628	-0,386	-0,056	1	0,751	0,447	0,738	0,507	0,718	0,681
LOC	0,628	0,326	0,681	-0,473	-0,150	0,751	1	0,467	0,905	0,654	0,968	0,714
NOA	0,412	0,382	0,445	-0,423	-0,154	0,447	0,467	1	0,316	0,126	0,510	0,212
NOO	0,421	0,383	0,468	-0,384	-0,153	0,738	0,905	0,316	1	0,619	0,892	0,822
RFC	0,553	0,124	0,531	0,079	-0,276	0,507	0,654	0,126	0,619	1	0,618	0,486
WMPC	0,599	0,355	0,630	-0,514	-0,191	0,718	0,968	0,510	0,892	0,618	1	0,695
WTcov	0,353	0,488	0,395	-0,372	-0,077	0,681	0,714	0,212	0,822	0,486	0,695	1

Tableau 25 - Corrélations de Spearman des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	-0,020	0,840	-0,250	-0,190	0,427	0,341	-0,138	0,112	0,468	0,389	0,233

COC	-0,020	1	-0,051	-0,378	0,592	0,420	0,023	-0,400	0,367	-0,065	-0,050	0,495
FO	0,840	-0,051	1	-0,287	-0,175	0,481	0,401	0,014	0,323	0,346	0,567	0,425
DOIH	-0,250	-0,378	-0,287	1	-0,429	-0,546	0,072	0,309	-0,468	0,182	-0,109	-0,539
NOCC	-0,190	0,592	-0,175	-0,429	1	0,307	-0,262	-0,405	0,337	-0,388	-0,203	0,514
LCOM	0,427	0,420	0,481	-0,546	0,307	1	0,772	0,199	0,644	0,792	0,666	0,838
LOC	0,341	0,023	0,401	0,072	-0,262	0,772	1	0,380	0,591	0,876	0,889	0,515
NOA	-0,138	-0,400	0,014	0,309	-0,405	0,199	0,380	1	0,088	0,218	0,453	-0,055
NOO	0,112	0,367	0,323	-0,468	0,337	0,644	0,591	0,088	1	0,312	0,668	0,887
RFC	0,468	-0,065	0,346	0,182	-0,388	0,792	0,876	0,218	0,312	1	0,694	0,294
WMPC	0,389	-0,050	0,567	-0,109	-0,203	0,666	0,889	0,453	0,668	0,694	1	0,592
WTcov	0,233	0,495	0,425	-0,539	0,514	0,838	0,515	-0,055	0,887	0,294	0,592	1

Tableau 26 - Corrélations de Spearman des métriques OO des classes de CAT-1

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,510	0,956	-0,379	-0,102	0,835	0,869	0,588	0,802	0,847	0,833	0,710
COC	0,510	1	0,507	-0,294	0,005	0,679	0,653	0,418	0,710	0,565	0,684	0,601
FO	0,956	0,507	1	-0,389	-0,094	0,816	0,833	0,589	0,805	0,807	0,804	0,722
DOIH	-0,379	-0,294	-0,389	1	0,087	-0,241	-0,371	-0,333	-0,339	-0,110	-0,359	-0,348
NOCC	-0,102	0,005	-0,094	0,087	1	-0,052	-0,069	-0,147	-0,054	-0,109	-0,076	-0,031
LCOM	0,835	0,679	0,816	-0,241	-0,052	1	0,915	0,406	0,955	0,892	0,916	0,815
LOC	0,869	0,653	0,833	-0,371	-0,069	0,915	1	0,558	0,952	0,902	0,988	0,812
NOA	0,588	0,418	0,589	-0,333	-0,147	0,406	0,558	1	0,435	0,438	0,530	0,301
NOO	0,802	0,710	0,805	-0,339	-0,054	0,955	0,952	0,435	1	0,880	0,948	0,860
RFC	0,847	0,565	0,807	-0,110	-0,109	0,892	0,902	0,438	0,880	1	0,889	0,724
WMPC	0,833	0,684	0,804	-0,359	-0,076	0,916	0,988	0,530	0,948	0,889	1	0,811
WTcov	0,710	0,601	0,722	-0,348	-0,031	0,815	0,812	0,301	0,860	0,724	0,811	1

Tableau 27 - Corrélations de Pearson des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,126	0,905	-0,231	-0,133	0,899	0,756	-0,237	0,778	0,809	0,769	0,860
COC	0,126	1	0,039	-0,335	0,739	0,586	0,026	-0,274	0,225	-0,028	-0,014	0,252
FO	0,905	0,039	1	-0,276	-0,180	0,718	0,755	-0,189	0,642	0,643	0,763	0,734
DOIH	-0,231	-0,335	-0,276	1	-0,384	-0,387	-0,065	0,082	-0,270	0,113	-0,157	-0,291
NOCC	-0,133	0,739	-0,180	-0,384	1	-0,039	-0,205	-0,276	-0,056	-0,297	-0,216	-0,025
LCOM	0,899	0,586	0,718	-0,387	-0,039	1	0,815	-0,130	0,947	0,921	0,852	0,975
LOC	0,756	0,026	0,755	-0,065	-0,205	0,815	1	0,202	0,830	0,813	0,944	0,802
NOA	-0,237	-0,274	-0,189	0,082	-0,276	-0,130	0,202	1	0,102	-0,010	0,220	-0,091
NOO	0,778	0,225	0,642	-0,270	-0,056	0,947	0,830	0,102	1	0,697	0,841	0,974
RFC	0,809	-0,028	0,643	0,113	-0,297	0,921	0,813	-0,010	0,697	1	0,738	0,702
WMPC	0,769	-0,014	0,763	-0,157	-0,216	0,852	0,944	0,220	0,841	0,738	1	0,823
WTcov	0,860	0,252	0,734	-0,291	-0,025	0,975	0,802	-0,091	0,974	0,702	0,823	1

Tableau 28 - Corrélations de Pearson des métriques OO des classes de CAT-1

Après avoir généré les données de corrélation de Spearman et de Pearson des métriques orientées-objet et du taux de couverture à l'aide de l'outil XLSTAT, passons à présent à l'analyse respective des *tableaux 25, 26, 27 et 28* :

- À première vue, il semblerait que les tableaux de corrélation entre les métriques orientées-objet et le taux de couverture présentent le même « pattern » pour les catégories CAT-2 et CAT-1: nous observons des corrélations significatives pour les métriques de couplage, de taille et de complexité. Par ailleurs, nous voyons que l'héritage (à travers DOIH) n'est pas corrélé au taux de couverture, pour chacune des catégories.
- La cohésion (LCOM) également est fortement corrélée au taux de couverture avec une plus forte intensité en faveur de CAT-1 (*tableaux 26 et 28*).

- Notons également qu'au niveau de la catégorie 1, la métrique NOA n'a pas une corrélation significative avec le taux de couverture, et cela peut se constater dans les statistiques descriptives, où la moyenne des valeurs de NOA est relativement plus faible dans la catégorie CAT-1 (*tableau 23*) comparée à celle trouvée dans la catégorie CAT-2 dans le *tableau 22* (1,36 contre 3,44). Ce facteur pourrait être déterminant quant au fait qu'une classe soit testée ou pas. Si nous nous penchons sur la métrique LOC, nous voyons bien que cette métrique est significativement corrélée au taux de couverture en ce qui concerne la catégorie CAT-2 (0,71 et 0,81 respectivement Pearson et Spearman des *tableaux 25 et 27*) ce qui n'est pas vraiment le cas de la catégorie CAT-1 (0,80 et 0,51 respectivement Pearson et Spearman des *tableaux 26 et 28*) - la métrique LOC pourrait donc aussi être déterminante dans la décision de tester une classe ou pas.

L'analyse des corrélations révèle un certain nombre d'informations supplémentaires quant aux caractéristiques des classes pour lesquelles des tests JUnit ont été développés, mais également pour celles pour lesquelles nous avons des données de taux de couverture non nulles alors qu'il n'y a pas eu de tests développés pour ces classes. En effet, les classes de CAT-2 se démarquent surtout par les métriques LOC et NOA, plus corrélées au taux de couverture que celles de CAT-1. Ces caractéristiques de taille pourraient être des facteurs dans la détermination des tests à développer.

Une caractéristique commune aux deux catégories, serait la forte corrélation positive de la métrique de complexité WMPC ainsi que de la métrique de cohésion LCOM avec le taux de couverture.

4.4.3.3. Analyse des courbes de tendances

En sélectionnant quelques métriques (COC, FO, DOIH, LOC et WMPC) à la suite de l'étape expérimentale précédente, tâchons d'analyser les tendances de variation des valeurs des métriques et du taux de couverture dans chaque catégorie.

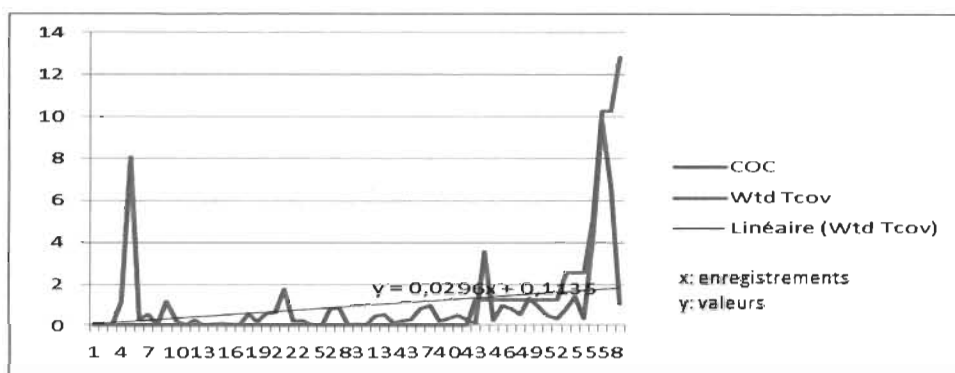


Figure 23 - Variation du taux de couverture vs la métrique COC (CAT-2)

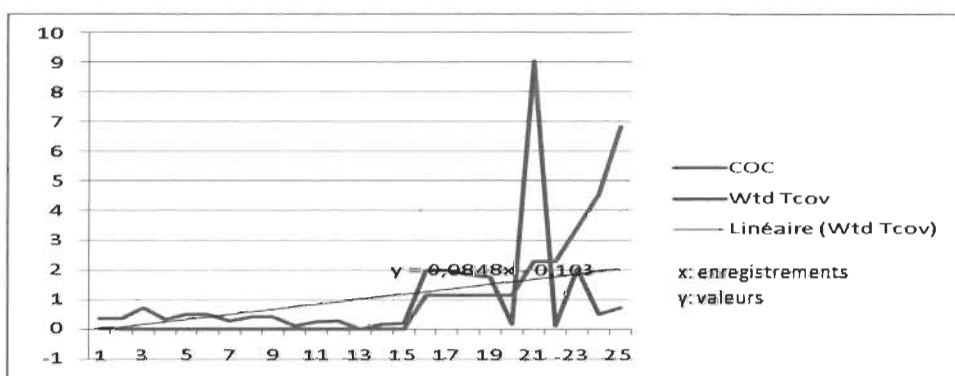


Figure 24 - Variation du taux de couverture vs la métrique COC (CAT-1)

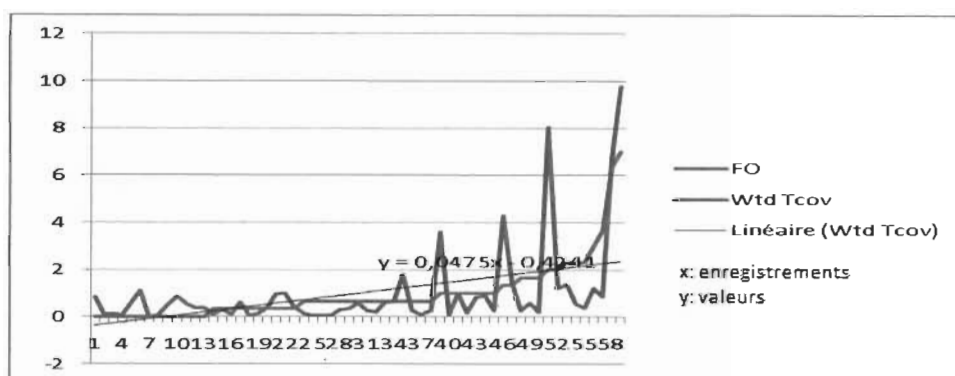


Figure 25 - Variation du taux de couverture vs la métrique FO (CAT-2)

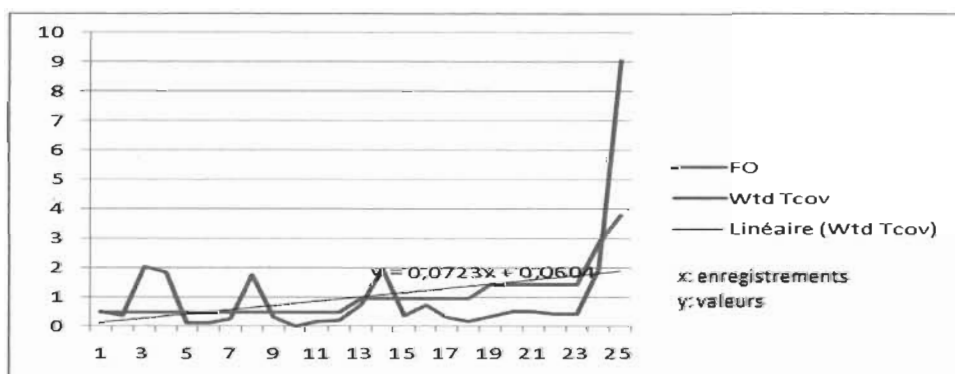


Figure 26 - Variation du taux de couverture vs la métrique FO (CAT-1)

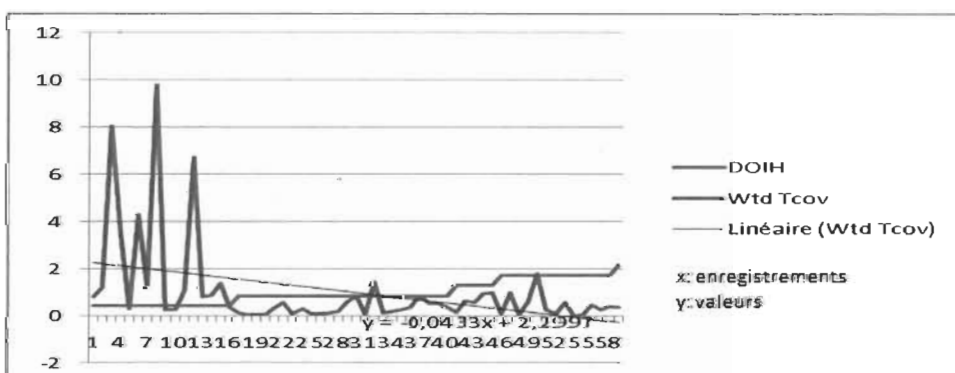


Figure 27 - Variation du taux de couverture vs la métrique DOIH (CAT-2)

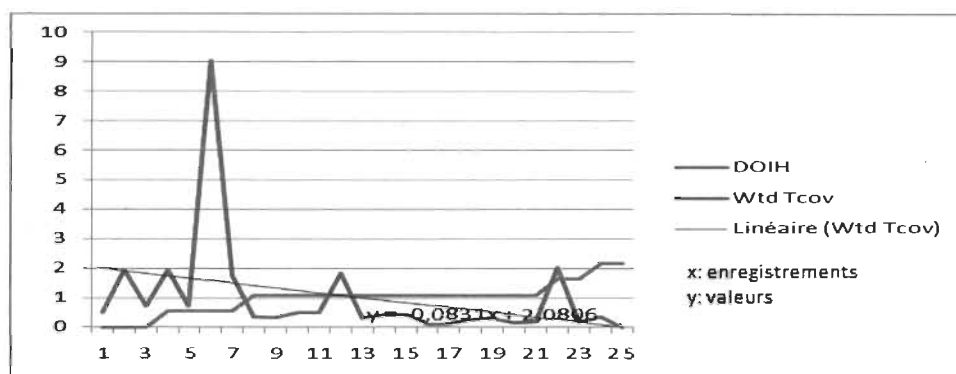


Figure 28 - Variation du taux de couverture vs la métrique DOIH (CAT-1)

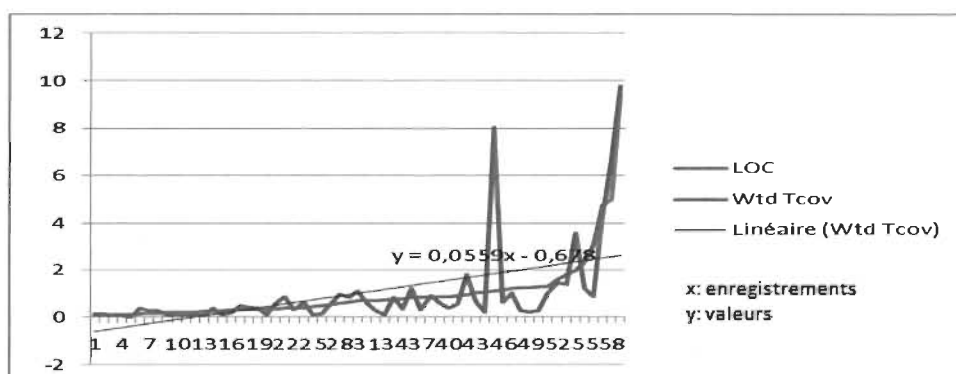


Figure 29 - Variation du taux de couverture vs la métrique LOC (CAT-2)

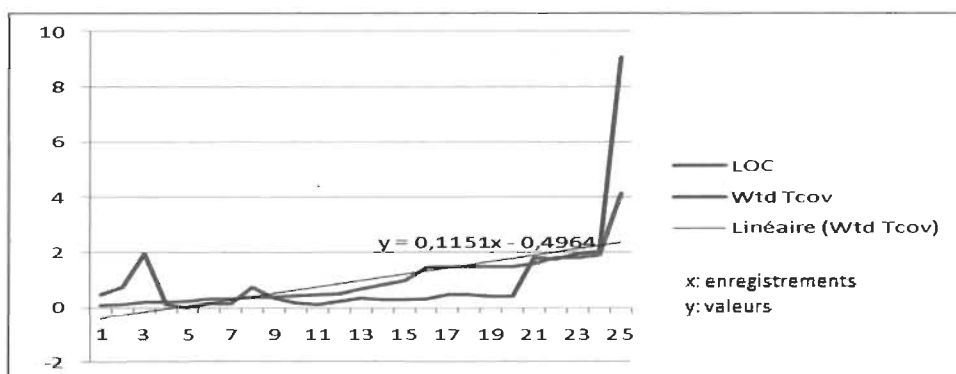


Figure 30 - Variation du taux de couverture vs la métrique LOC (CAT-1)

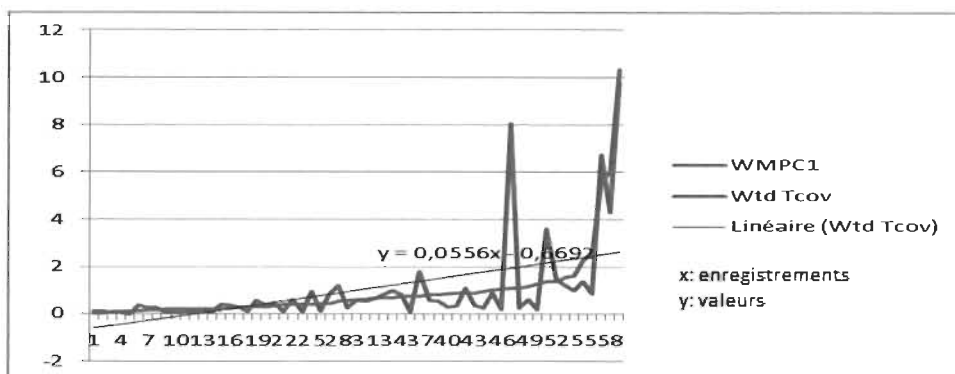


Figure 31 - Variation du taux de couverture vs la métrique WMPC (CAT-2)

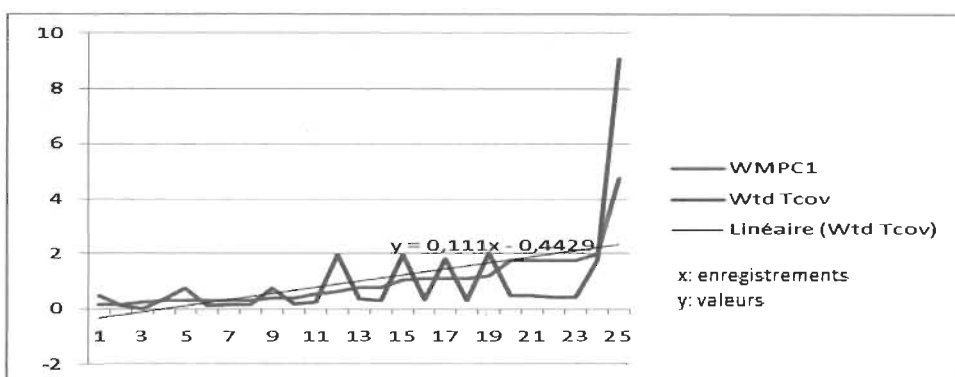


Figure 32 - Variation du taux de couverture vs la métrique WMPC (CAT-1)

Les 10 graphes de tendances précédents s'intègrent dans notre démarche expérimentale comme des éléments qui vont permettre de confirmer ou d'infirmer nos premiers constats basés sur l'analyse des statistiques descriptives et des corrélations:

- L'analyse des courbes de tendances montrent que: en effet, il semblerait que les métriques COC, FO, LOC et WMPC caractérisent les classes appartenant aux catégories CAT-2 et CAT-1 lorsque nous voyons que ces métriques influencent positivement sur la tendance de l'étendue de la couverture des tests JUnit tel que nous le révèlent les figures 23, 24,

25, 26, 29, 30, 31 et 32. Les pentes des droites de tendance de l'évolution du taux de couverture en fonction de l'évolution des valeurs de ces dernières métriques sont positives: ces pentes varient de 0,029 (pour COC versus le taux de couverture pour les classes de CAT-2, *figure 23*) à 0,1151 (pour LOC versus le taux de couverture pour les classes de CAT-1, *figure 30*); notons par ailleurs que les pentes des courbes de tendances de l'évolution du taux de couverture en fonction des métriques COC, FO, WMPC et LOC pour CAT-1 sont supérieures à celles pour CAT-2: cela pourrait donc expliquer le fait que les classes de la catégorie CAT-1 aient été touchées par les tests JUnit développées lors de l'exécution de ces tests qui à l'origine étaient développés pour les classes de la catégorie CAT-2. Il y aurait un effet de propagation des tests du fait des caractéristiques des classes de CAT-1 quant à leur couplage, leur taille et leur complexité.

- Les *figures 27 et 28* montrent également bien comment l'héritage à travers la métrique DOIH n'exerce aucune influence sur le taux de couverture étant donné que l'équation de la droite de tendance est négative autant pour CAT-2 que pour CAT-1.

Les caractéristiques qui se dégagent de nos expérimentations jusqu'ici montrent la forte et significative corrélation positive entre les métriques de couplage (COC), de taille (LOC) et de complexité (WMPC) et le taux de couverture, tandis que l'héritage (DOIH) ne semble pas affecter le taux de couverture.

4.4.3.4. Classification K-Means

À présent que nous avons identifié une série de métriques pouvant constituer des caractéristiques des catégories CAT-2 et CAT-1 du fait de leur forte corrélation avec le taux de couverture et qu'également nous avons noté que la métrique d'héritage DOIH caractérisait les classes de la catégorie CAT-0, appliquons une classification K-Means avec $k=3$ sur l'ensemble des données.

Les données de métriques que nous considérerons, suite aux résultats des étapes précédentes de nos expérimentations, seront: FO – COC – LOC – WMPC – DOIH.

Voici les résultats obtenus présentés dans le *tableau 29*:

Attribute	Full Data (109)	0 (7)	1 (25)	2 (77)
COC	0.7982	6.1429	0.36	0.4545
FO	2.7064	8.4286	1.96	2.4286
DOIH	2.0826	1	3.76	1.6364
LOC	110.7431	336.8571	155.72	75.5844
WMPC	18.8716	91.8571	12.36	14.3506
Wtd_Icov	546.3028	4103.7143	359.32	283.6104

Tableau 29 - Synthèse de la classification k-means

Après mise en correspondance des catégories et des clusters nous obtenons le *tableau 30* d'affectation suivant:

	CAT-2	CAT-1	CAT-0	Total
Cluster2	35	20	22	77
Cluster1	19	4	2	25
Cluster0	5	1	1	7
Total	59	25	25	109

Tableau 30 - Correspondance des catégories et les clusters

L'analyse de ces données nous aidera à vérifier nos précédents constats:

- D'après le *tableau 29*, la classification K-Means avec $K = 3$, nous propose trois clusters avec les caractéristiques suivantes:
 - Cluster2: comprend 77 classes (45% CAT-2 ; 26% CAT-1 et 29% CAT-0) pour lesquelles les valeurs de LOC (métrique liée à la taille) sont les plus faibles (75,58). Par ailleurs, ce n'est pas tant la faible valeur du FAN-In qui est significative (0,45), mais plus le rapport entre le FAN-Out et le FAN-In - les classes appartenant à ce cluster font nettement plus d'appels sortants qu'elles n'ont d'appels entrants (environ 5 fois plus).
 - Cluster1: 25 classes (76% CAT-2, 16% CAT-1 et 8% CAT-0), qui se caractérisent essentiellement par un DOIH élevé (3,76) et les plus faibles valeurs du couplage (COC et FO), de la complexité (WMPC).
 - Cluster0: 7 classes (72% CAT-2, 14% CAT-1 et 14% CAT-0) - on y trouve des valeurs élevées de presque toutes les métriques. Cela démontre bien que les classes de la catégorie CAT-2 qui forment essentiellement ce cluster sont des classes ayant une complexité particulière (WMPC = 91,85), une taille conséquente (LOC = 336,85), un couplage conséquent (FO :

8,42, COC : 6,14) et bien entendu, un taux de couverture élevé (4103, 71).

- Concernant la répartition des catégories de classes:
 - CAT-2: (59% des classes de cette catégorie se retrouvent dans le Cluster2, 32% dans le Cluster1 et 9% dans le Cluster0).
 - CAT-1: (80% cluster2, 16% cluster1 et 4% cluster0) - plus de 80% des classes de cette catégorie sont retrouvées dans le cluster2, cluster caractérisé par des valeurs de FAN-In nettement inférieures aux valeurs de FAN-Out.
 - CAT-0: (88% cluster2, 8% cluster1 et 4% cluster0) - en réalité, 1 seule classe de cette catégorie est retrouvée dans le cluster0 où l'on retrouve des valeurs élevées de toutes les métriques sélectionnées (COC, FO, LOC, WMPC, WTcov) à l'exception de DOIH, confortant ainsi l'hypothèse selon laquelle les classes de la catégorie CAT-0 sont des classes ayant de faibles valeurs de métriques à l'exception des métriques d'héritage (en l'occurrence DOIH).

L'analyse par classification K-Means fournit des résultats partiellement en accord avec les expérimentations des étapes antérieures, à savoir:

- la complexité, la taille et le couplage élevés sont des caractéristiques des classes de CAT-2,
- mais les classes de la catégorie CAT-1 ne dénotent pas un FAN-In plus important que le FAN-Out, ce qui diffère de nos précédents constats.

En effet, ceci peut s'expliquer à travers les statistiques descriptives des classes du système IO en général (comme nous avons pu le constater au niveau des *tableaux 22, 23 et 24*). Le système IO se caractérise dans sa globalité par des valeurs plus faibles du FAN-In par rapport au FAN-Out,

- les classes de la catégorie CAT-0, outre des valeurs plus faibles quant aux métriques de complexité et de taille, possèdent des valeurs de métriques DOIH (héritage) non négligeables.

4.4.4 POI

4.4.4.1. Analyse des statistiques descriptives

POI est le quatrième système que nous étudierons dans le cadre de ce chapitre. Tel que précédemment, procédons tout d'abord à une analyse des statistiques descriptives de ses données de métriques, pour chacune des catégories CAT-2, CAT-1 et CAT-0. Cela va permettre de dégager quelques éléments caractéristiques de chaque système.

Variable	Observation	Minimum	Maximum	Moyenne
CBO	359	0,000	111,000	11,150
COC	359	0,000	200,000	9,114
FO	359	0,000	62,000	7,078
DOIH	359	0,000	5,000	2,192
NOCC	359	0,000	49,000	0,476
LCOM	321	0,000	10477,000	198,511
LOC	359	3,000	1427,000	151,574
NOA	359	0,000	20,000	3,457
NOO	359	0,000	188,000	15,708
RFC	359	1,000	642,000	64,838
WMPC	359	1,000	374,000	30,677
WTcov	359	0,000	14288,000	1062,253

Tableau 31 - Statistiques descriptives des données des classes de CAT-2

Variable	Observation	Minimum	Maximum	Moyenne
CBO	574	0,000	72,000	5,713
COC	574	0,000	199,000	5,645
FO	574	0,000	49,000	3,587
DOIH	574	0,000	6,000	1,840
NOCC	574	0,000	189,000	2,662
LCOM	426	0,000	37937,000	198,023
LOC	574	3,000	1631,000	70,446
NOA	574	0,000	145,000	2,641
NOO	574	0,000	294,000	8,585
RFC	574	1,000	323,000	38,073
WMPC	574	1,000	295,000	13,941
WTcov	574	0,000	6909,000	399,958

Tableau 32 - Statistiques descriptives des données des classes de CAT-1

Variable	Observation	Minimum	Maximum	Moyenne
CBO	606	0,000	168,000	4,249
COC	606	0,000	75,000	1,521
FO	606	0,000	28,000	2,490
DOIH	606	0,000	7,000	1,835
NOCC	606	0,000	38,000	0,865
LCOM	357	0,000	37953,000	196,006
LOC	606	2,000	1686,000	62,036
NOA	606	0,000	145,000	2,389
NOO	606	0,000	294,000	4,946
RFC	606	0,000	570,000	37,812
WMPC	606	0,000	295,000	9,180

Tableau 33 - Statistiques descriptives des données des classes de CAT-0

L'analyse des *tableaux 31, 32, 33* fait ressortir un certain nombre de points à savoir:

- POI compte 1539 classes dont près du quart (23%) appartiennent à la catégorie CAT-2, 37% à la catégorie CAT-1 et 40 % à la catégorie CAT-0. Ainsi, les tests JUnit n'ont pas été développés pour 77% des classes (soit 1180 classes) du système POI;
- Les classes de la catégorie CAT-2 présentent les valeurs les plus élevées pour ce qui est de la complexité (WMPC et RFC: 31 et 64 en moyenne, tel que le montre le *tableau 31*), la taille (LOC, NOA et NOO), et le couplage (CBO, FO, COC). Rappelons qu'il s'agit ici des classes pour lesquelles des tests JUnit ont été développés et dont bien entendu nous disposons de valeurs du taux de couverture pondéré non nulles voire même relativement élevées (moyenne: 1062) comparativement aux valeurs moyennes trouvées au niveau des de la catégorie CAT-1 (400, *tableau 32*). En partant de la formule de calcul du taux de couverture

pondéré (cf. *Chapitre 3*) et en notant que le nombre d'opérations (métrique NOO) en moyenne pour chaque classe de la CAT-2 est de 15,71, on peut avancer qu'en moyenne 67,61% ($1062,25 / 15,71$) des méthodes de chaque classe sont appelées lors des tests;

- Concernant les classes de la catégorie CAT-1 et CAT-0 (*tableaux 32 et 33*), celles-ci présentent plus ou moins les mêmes caractéristiques si l'on se base sur les données des métriques au niveau des statistiques descriptives. Néanmoins, les classes de la catégorie CAT-1 présentent une moyenne de couplage entrant plus élevé que le couplage sortant ($COC = 5,645$ versus $FO = 3,587$) par rapport aux données des classes de CAT-0 ($COC = 1,521$ versus $FO = 2,490$), ainsi qu'une moyenne plus élevée de la métrique NOO (pour CAT-2 $NOO = 8,585$ versus pour CAT-1 $NOO = 4,946$). Ces dernières caractéristiques des classes de CAT-1 pourraient expliquer la raison pour laquelle ces classes soient atteintes par les tests JUnit développées pour les classes de la catégorie CAT-2 et qu'aucun test JUnit n'a été développé pour elles. Ces caractéristiques montrent que leur couplage entrant est relativement fort et significatif comparé à leur couplage sortant, et que leur nombre d'opérations est aussi significatif.

Ainsi, bien qu'aucun test JUnit n'ait été développé pour les classes de la catégorie CAT-1, 46,58% ($399,958 / 8,585$), des méthodes de chacune de ces classes ont été atteintes par les tests JUnit développés pour les classes de la catégorie CAT-2.

- En d'autres termes, les tests JUnit développés pour les classes de la catégorie CAT-2 auraient permis de couvrir à 100%:
 - 242 classes sur 359 pour la catégorie CAT-2 (67,61 % calculé plus tôt),
 - et 267 classes sur 574 pour la catégorie CAT-1 (46,58% calculé plus tôt).

Nous pourrions donc prétendre que si des tests JUnit avaient été développés pour couvrir toutes les méthodes de 242 classes sur 1539 classes (16%) que compte le système POI, alors cela aurait permis de couvrir en tout 509 classes (33% du système), par l'effet de propagation dû au fort couplage des classes des catégories CAT-2 et CAT-1, soit environ 2 fois plus de classes.

L'analyse des statistiques descriptives du système fait clairement ressortir que les classes de la catégorie CAT-2 se distinguent par des valeurs élevées des métriques de taille, de complexité et de couplage (nous laisserons de côté les métriques de cohésion du fait du manque de données non négligeable). Les classes de la catégorie CAT-1 ont un couplage entrant (COC) plus élevé que le couplage sortant (FO) contrairement aux classes de la catégorie CAT-0, et leurs nombres de méthodes (NOO) sont significativement plus élevés que ceux de la catégorie CAT-0 (presque 2 fois plus).

Ainsi, les familles de métriques à suivre de près seraient la taille, la complexité et le couplage. Les métriques de cohésion étant biaisées par le fait qu'il nous manque considérablement de données et les métriques d'héritage ne semblant pas être une caractéristique particulière distinguant les classes entre elles.

4.4.4.2. Analyse des corrélations (Spearman et Pearson)

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,185	0,937	-0,166	0,039	0,271	0,571	0,291	0,325	0,628	0,533	0,274
COC	0,185	1	0,181	-0,076	0,245	0,306	0,370	0,272	0,501	0,386	0,429	0,517
FO	0,937	0,181	1	-0,177	0,026	0,301	0,560	0,279	0,344	0,615	0,545	0,293
DOIH	-0,166	-0,076	-0,177	1	-0,228	0,039	-0,083	0,191	0,076	0,223	-0,128	-0,003
NOCC	0,039	0,245	0,026	-0,228	1	0,169	0,011	-0,109	0,071	0,005	0,053	0,061
LCOM	0,271	0,306	0,301	0,039	0,169	0,533	0,417	1	0,655	0,515	0,558	0,583
LOC	0,571	0,370	0,560	-0,083	0,011	0,417	1	0,519	0,765	0,726	0,918	0,706
NOA	0,291	0,272	0,279	0,191	-0,109	1	0,519	1	0,572	0,450	0,505	0,512
NOO	0,325	0,501	0,344	0,076	0,071	0,655	0,765	0,572	1	0,712	0,842	0,929
RFC	0,628	0,386	0,615	0,223	0,005	0,515	0,726	0,450	0,712	1	0,722	0,616
WMPC	0,533	0,429	0,545	-0,128	0,053	0,558	0,918	0,505	0,842	0,722	1	0,791
WTcov	0,274	0,517	0,293	-0,003	0,061	0,583	0,706	0,512	0,929	0,616	0,791	1

Tableau 34 - Corrélations de Spearman des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,143	0,937	0,048	-0,119	0,242	0,657	0,319	0,437	0,625	0,599	0,343
COC	0,143	1	0,131	-0,167	0,342	0,345	0,283	0,159	0,445	0,128	0,363	0,537
FO	0,937	0,131	1	0,061	-0,078	0,278	0,558	0,264	0,400	0,588	0,533	0,314
DOIH	0,048	-0,167	0,061	1	-0,353	0,204	0,129	0,046	-0,009	0,497	0,040	-0,242
NOCC	-0,119	0,342	-0,078	-0,353	1	0,277	-0,206	-0,221	0,071	-0,209	-0,055	0,204
LCOM	0,242	0,345	0,278	0,204	0,277	1	0,433	0,141	0,563	0,502	0,511	0,427
LOC	0,657	0,283	0,558	0,129	-0,206	0,433	1	0,561	0,766	0,757	0,902	0,566
NOA	0,319	0,159	0,264	0,046	-0,221	0,141	0,561	1	0,471	0,342	0,492	0,352
NOO	0,437	0,445	0,400	-0,009	0,071	0,563	0,766	0,471	1	0,607	0,896	0,780
RFC	0,625	0,128	0,588	0,497	-0,209	0,502	0,757	0,342	0,607	1	0,706	0,347
WMPC	0,599	0,363	0,533	0,040	-0,055	0,511	0,902	0,492	0,896	0,706	1	0,703
WTcov	0,343	0,537	0,314	-0,242	0,204	0,427	0,566	0,352	0,780	0,347	0,703	1

Tableau 35 - Corrélations de Spearman des métriques OO des classes de CAT-1

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,220	0,954	-0,148	0,153	0,599	0,733	0,352	0,642	0,822	0,742	0,563
COC	0,220	1	0,234	-0,176	0,081	0,157	0,277	0,078	0,307	0,267	0,319	0,314
FO	0,954	0,234	1	-0,159	0,011	0,605	0,679	0,366	0,663	0,822	0,728	0,553
DOIH	-0,148	-0,176	-0,159	1	-0,117	-0,078	-0,099	0,159	-0,029	0,013	-0,145	-0,057
NOCC	0,153	0,081	0,011	-0,117	1	0,076	0,257	-0,056	0,029	0,029	0,025	0,028
LCOM	0,599	0,157	0,605	-0,078	0,076	1	0,674	0,314	0,838	0,757	0,767	0,763
LOC	0,733	0,277	0,679	-0,099	0,257	0,674	1	0,517	0,783	0,782	0,891	0,751
NOA	0,352	0,078	0,366	0,159	-0,056	0,314	0,517	1	0,508	0,409	0,492	0,468
NOO	0,642	0,307	0,663	-0,029	0,029	0,838	0,783	0,508	1	0,853	0,899	0,935
RFC	0,822	0,267	0,822	0,013	0,029	0,757	0,782	0,409	0,853	1	0,858	0,766
WMPC	0,742	0,319	0,728	-0,145	0,025	0,767	0,891	0,492	0,899	0,858	1	0,877
WTcov	0,563	0,314	0,553	-0,057	0,028	0,763	0,751	0,468	0,935	0,766	0,877	1

Tableau 36 - Corrélations de Pearson des métriques OO des classes de CAT-2

Variable	CBO	COC	FO	DOIH	NOCC	LCOM	LOC	NOA	NOO	RFC	WMPC	WTcov
CBO	1	0,150	0,877	-0,066	-0,007	-0,005	0,432	0,071	0,147	0,518	0,402	0,292
COC	0,150	1	0,141	-0,153	0,310	0,005	0,123	0,005	0,084	0,057	0,139	0,236
FO	0,877	0,141	1	-0,050	-0,042	-0,005	0,406	0,090	0,163	0,530	0,400	0,318
DOIH	-0,066	-0,153	-0,050	1	-0,091	-0,058	-0,033	-0,059	-0,047	0,238	-0,060	-0,117
NOCC	-0,007	0,310	-0,042	-0,091	1	0,006	-0,047	-0,037	-0,025	-0,072	-0,035	0,004
LCOM	-0,005	0,005	-0,005	-0,058	0,006	1	0,741	0,909	0,857	0,483	0,737	0,204
LOC	0,432	0,123	0,406	-0,033	-0,047	0,741	1	0,759	0,855	0,739	0,932	0,506
NOA	0,071	0,005	0,090	-0,059	-0,037	0,909	0,759	1	0,876	0,482	0,769	0,316
NOO	0,147	0,084	0,163	-0,047	-0,025	0,857	0,855	0,876	1	0,670	0,920	0,554
RFC	0,518	0,057	0,530	0,238	-0,072	0,483	0,739	0,482	0,670	1	0,759	0,446
WMPC	0,402	0,139	0,400	-0,060	-0,035	0,737	0,932	0,769	0,920	0,759	1	0,620
WTcov	0,292	0,236	0,318	-0,117	0,004	0,204	0,506	0,316	0,554	0,446	0,620	1

Tableau 37 - Corrélations de Pearson des métriques OO des classes de CAT-1

Après avoir généré les données de corrélation de Spearman et de Pearson des métriques orientées-objet et du taux de couverture à l'aide de l'outil XLSTAT, nous passons à présent à l'analyse respective des *tableaux 34, 35, 36 et 37* :

- De prime abord, nous pouvons noter que l'héritage (DOIH et NOCC) et le taux de couverture ne présentent vraisemblablement pas de corrélation quelconque.
- Par ailleurs, dans les *tableaux 34 et 35*, autant pour les données relatives aux classes de la catégorie CAT-2 que pour celles relatives aux classes de la catégorie CAT-1, nous voyons que le couplage entrant (COC) est plus fortement et plus significativement corrélé au taux de couverture que le couplage sortant (FO): dans CAT-2, nous avons 0,517 contre 0,293; et dans CAT-1 nous avons 0,537 versus 0,314 (Spearman).
- Enfin, d'une manière générale, les classes de la catégorie CAT-2 sont plus significativement et plus fortement corrélées au taux de couverture, pour ce qui est des métriques de taille (LOC en particulier) et de complexité (WMPC en particulier), comme nous le montrent les *tableaux 34 et 36*.

L'analyse des corrélations révèle un certain nombre d'informations supplémentaires quant aux caractéristiques des classes pour lesquelles des tests JUnit ont été développés, mais également pour celles pour lesquelles nous avons des données de métrique du taux de couverture non nulles alors qu'il n'y a pas eu de tests développées pour ces classes. En effet, les classes de CAT-2 et CAT-1 se ressemblent du point de vue du rapport de leur couplage entrant qui est plus fort que leur couplage sortant. Ceci est dû à leur importante taille et de leur importante complexité, les classes de la catégorie CAT-2 se distinguent nettement de celles de CAT-1.

4.4.4.3. Analyse des courbes de tendances

En sélectionnant quelques métriques (COC, FO, DOIH, LOC et WMPC) suite aux résultats obtenus lors de l'étape expérimentale précédente, tâchons d'analyser les tendances de variation des valeurs des métriques et du taux de couverture dans chaque catégorie.

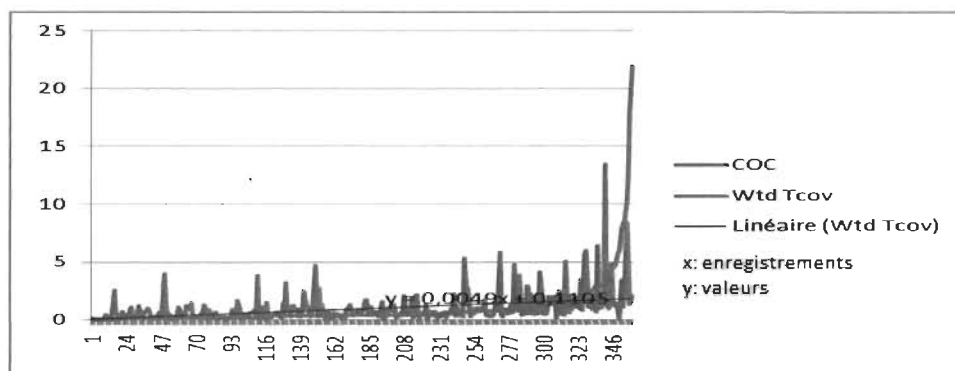


Figure 33 - Variation du taux de couverture vs la métrique COC (CAT-2)

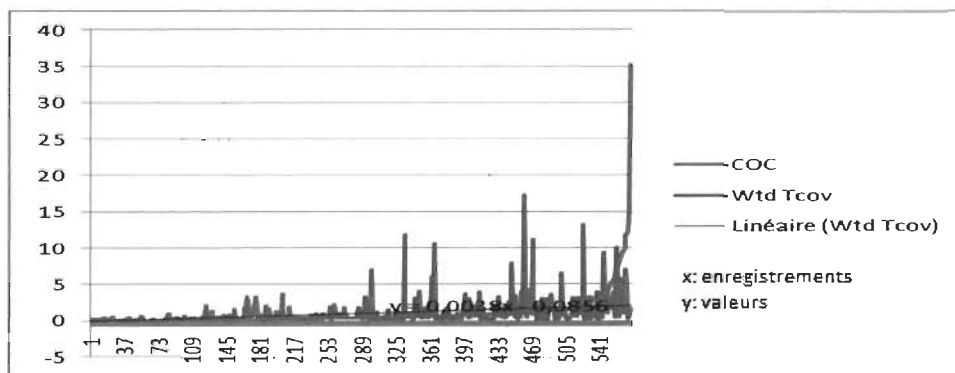


Figure 34 - Variation du taux de couverture vs la métrique COC (CAT-1)

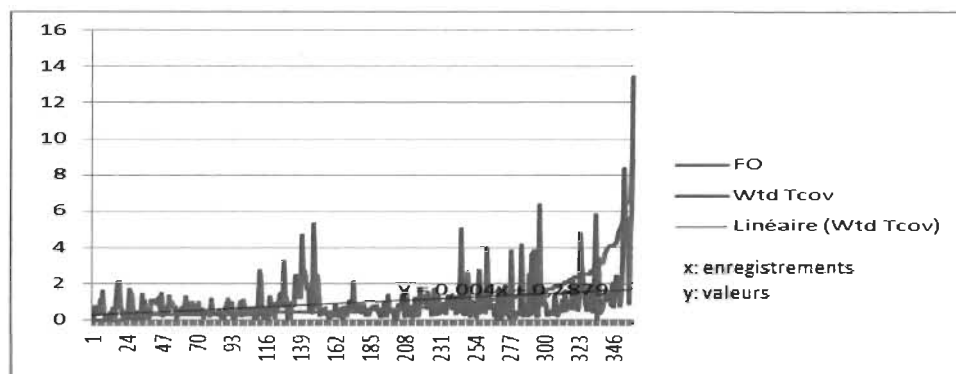


Figure 35 - Variation du taux de couverture vs la métrique FO (CAT-2)

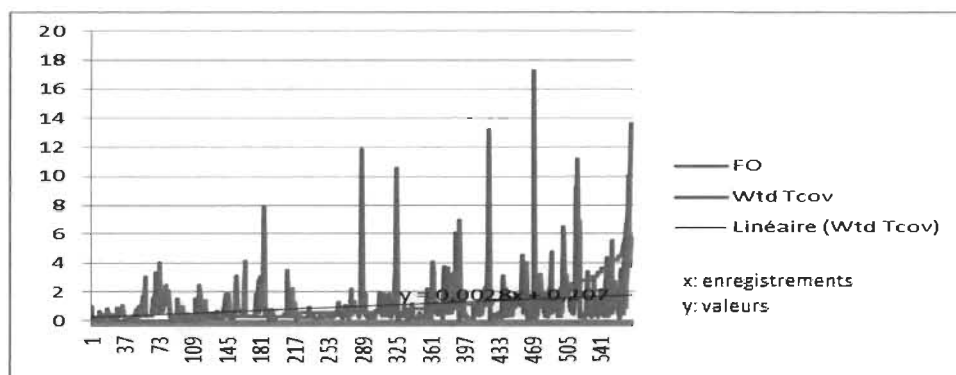


Figure 36 - Variation du taux de couverture vs la métrique FO (CAT-1)

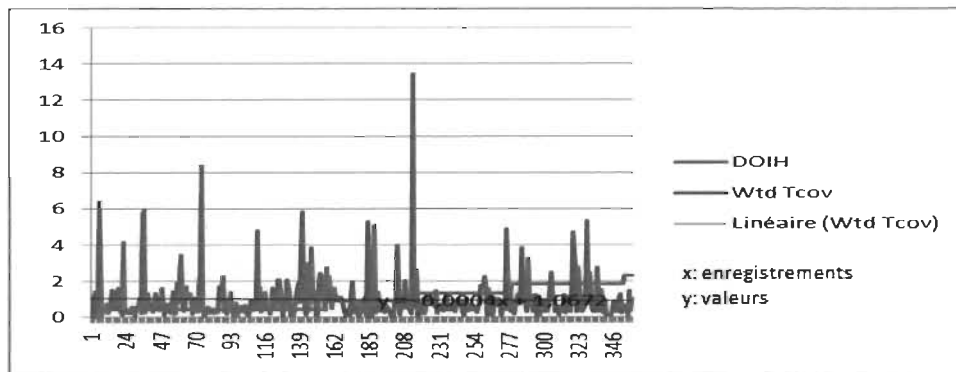


Figure 37 - Variation du taux de couverture vs la métrique DOIH (CAT-2)

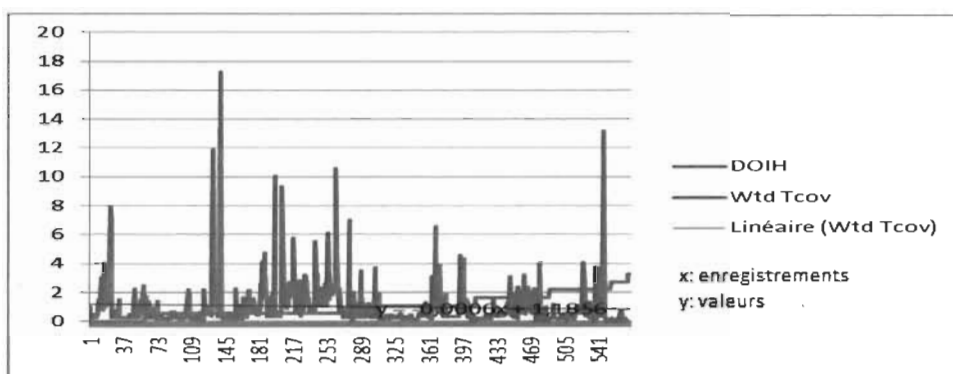


Figure 38 - Variation du taux de couverture vs la métrique DOIH (CAT-1)

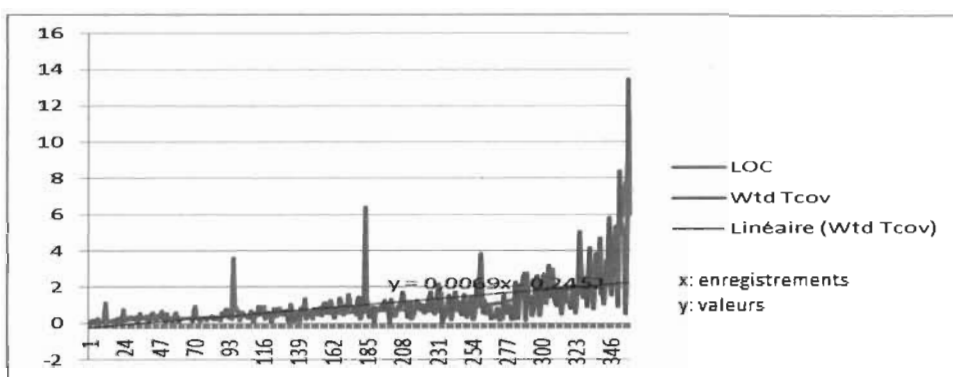


Figure 39 - Variation du taux de couverture vs la métrique LOC (CAT-2)

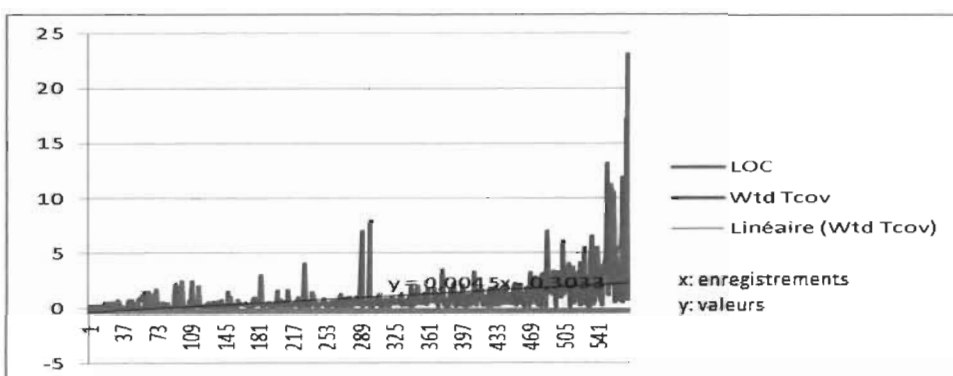


Figure 40 - Variation du taux de couverture vs la métrique LOC (CAT-1)

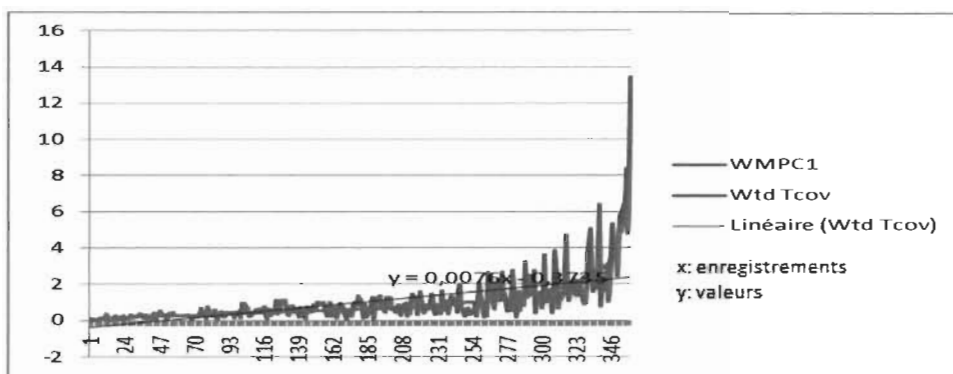


Figure 41 - Variation du taux de couverture vs la métrique WMPC (CAT-2)

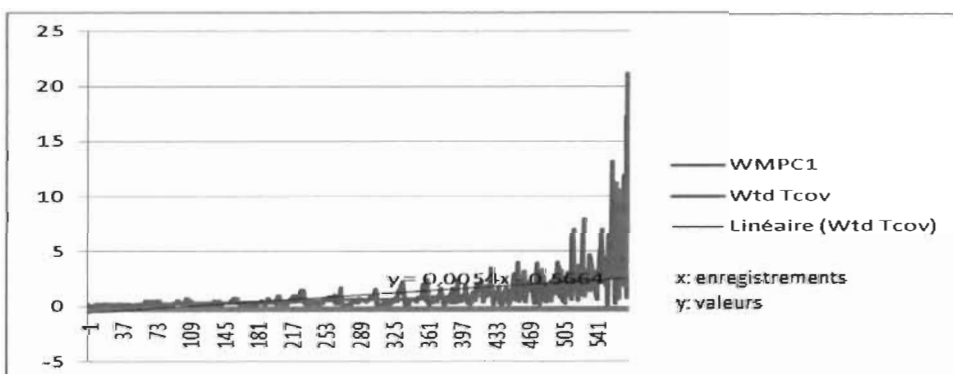


Figure 42 - Variation du taux de couverture vs la métrique WMPC (CAT-1)

Les figures 33 à 42 s'intègrent dans notre démarche expérimentale comme des éléments qui vont permettre de confirmer ou d'infirmer nos premiers constats basés sur l'analyse des statistiques descriptives et des corrélations:

- L'analyse des courbes des figures 39, 40, 41 et 42 montrent que: en effet, il semblerait que les métriques LOC et WMPC caractérisent les classes appartenant aux catégories CAT-2 et CAT-1 lorsque nous voyons que ces métriques influencent positivement sur la tendance de l'étendue de la couverture des tests JUnit. Les pentes des droites de l'évolution du taux de couverture en fonction de l'évolution des valeurs

de ces dernières métriques sont positives: ces pentes varient de 0,0045 (pour LOC versus le taux de couverture pour les classes de CAT-1) à 0,0076 (pour WMPC versus le taux de couverture pour les classes de CAT-2).

- Notons par ailleurs que les pentes des courbes de tendances de l'évolution du taux de couverture en fonction des métriques COC, FO, WMPC et LOC pour CAT-2 (*figures 33, 35, 39 et 41*) sont supérieures à celles pour CAT-1 (*figures 34, 36, 40 et 42*) - cela pourrait donc expliquer le fait que des tests JUnit aient été développés pour les classes de la catégorie CAT-2 et non pour les classes de CAT-1.
- Les pentes des droites de tendance de l'évolution du taux de couverture par rapport aux données de la métrique COC sont relativement fortes comparées aux pentes des droites de tendance de l'évolution du taux de couverture par rapport aux données de la métrique FO (0,0049 versus 0,004 pour CAT-2 et 0,0038 versus 0,0028 pour CAT-1); remarquons tout de même que les pentes des droites relatives aux classes de CAT-2 sont toujours supérieures à celles des droites relatives aux classes de CAT-1.
- Les *figures 37 et 38* montrent par ailleurs que l'héritage, à travers la métrique DOIH, semble n'exercer aucune influence sur le taux de couverture, étant donné que la pente de la droite de tendance est négative et tend vers 0 (autant pour CAT-2 que pour CAT-1).

Les caractéristiques qui se dégagent de nos expérimentations jusqu'ici montrent la forte et significative corrélation positive entre les métriques de couplage (COC), de taille (LOC) et de complexité (WMPC) et le taux de couverture, tandis que l'héritage (DOIH) ne semble pas affecter le taux de couverture.

4.4.4.4. Classification K-Means

À présent que nous avons identifié une série de métriques pouvant constituer des caractéristiques des catégories CAT-2 et CAT-1 du fait de leur forte corrélation avec le taux de couverture et qu'également nous avons noté que la métrique d'héritage DOIH caractérisait les classes de la catégorie CAT-0, appliquons une classification K-Means avec $k=3$ sur l'ensemble des données.

Les données de métriques que nous considérerons, suite aux résultats des étapes précédentes de nos expérimentations, seront: FO – COC – LOC – WMPC – DOIH.

Attribute	Full Data (1539)	Cluster#		
		0 (1024)	1 (433)	2 (82)
COC	4.8304	4.1992	3.0508	22.1098
FO	3.9695	3.1289	3.1016	19.0488
DOIH	1.9201	1.1953	3.7182	1.4756
LOC	86.0591	59.9072	66.3995	516.4512
WMPC	15.9701	11.2178	12.3279	94.5488
Wtd_Icov	616.9363	380.3906	478.0185	4304.4268

Tableau 38 - Synthèse de la classification k-Means

Après mise en correspondance des catégories et des clusters (*tableau 38*) nous obtenons le *tableau 39* suivant:

	CAT-2	CAT-1	CAT-0	Total
Cluster2	47	24	11	82
Cluster1	134	167	132	433
Cluster0	178	383	463	1024
Total	359	574	606	1539

Tableau 39 - Correspondance des catégories et les clusters

L'analyse de ces données nous aidera à vérifier nos précédents constats:

- D'après le *tableau 39*, la classification K-Means avec $K = 3$, nous propose trois clusters avec les caractéristiques suivantes:
 - Cluster2: comprend 82 classes (57% CAT-2; 30% CAT-1 et 13% CAT-0) pour lesquelles les valeurs de toutes les métriques sont les plus élevées à l'exception de celles de DOIH (1,4756). Par ailleurs, pour les classes de ce cluster, les valeurs du FAN-In (COC ou couplage entrant), sont supérieures à celles du FAN-Out (FO ou couplage sortant), à savoir 22,10 versus 19,04.
 - Cluster1: 433 classes (31% CAT-2, 39% CAT-1 et 30% CAT-0), qui se caractérisent essentiellement par un DOIH élevé (3,71) et les plus faibles valeurs du couplage (COC et FO).
 - Cluster0: 1024 classes (18% CAT-2, 37% CAT-1 et 45% CAT-0) - on y trouve les valeurs les plus faibles de presque toutes les métriques et près de la moitié des classes retrouvées dans ce cluster appartiennent à la catégorie CAT-0.

- Concernant la répartition des catégories de classes:

- CAT-2: (13% des classes de cette catégorie se retrouvent dans le Cluster2, 37% dans le Cluster1 et 50% dans le Cluster0).
- CAT-1: (4% cluster2, 29% cluster1 et 67% cluster0) - plus de 60% des classes de cette catégorie sont retrouvées dans le cluster0 où nous retrouvons le plus faible héritage (DOIH = 1,19) et où le Fan-In est plus élevé que le Fan-Out (4,19 versus 3,12); le cluster0 se caractérise entre autres par des valeurs de la métrique FAN-In nettement inférieures aux valeurs de la métrique FAN-Out. Ce constat vient confirmer l'hypothèse selon laquelle ces classes (de la catégorie CAT-1) sont couvertes par les tests JUnit, car elles possèdent des appels entrants conséquents.
- CAT-0: (2% cluster2, 22% cluster1 et 76% cluster0) - le cluster0 regroupant les classes pour lesquelles nous avons les valeurs les plus faibles pour toutes les métriques, cela conforte l'hypothèse selon laquelle les classes de la catégorie CAT-0 sont des classes ayant de faibles valeurs de métriques.

L'analyse par classification K-Means fournit des résultats en total accord avec les expérimentations des étapes antérieures, à savoir:

- la complexité, la taille et le couplage élevés sont des caractéristiques des classes de CAT-2,
- les classes de la catégorie CAT-1 dénotent un FAN-In plus important que le FAN-Out,

- les classes de la catégorie CAT-0 possèdent des valeurs plus faibles quant aux différentes métriques de complexité, de taille, de couplage et de taille.

CHAPITRE 5. OBSERVATION DE L'EFFORT DE TEST VERSUS LA COUVERTURE DE TEST

5.1. Objectif

Après avoir observé que la taille, la complexité ainsi que le couplage sont des caractéristiques apparentes des classes selon que celles-ci appartiennent aux catégories CAT-2, CAT-1 ou CAT-0, catégories définies en fonction du développement ou non des tests JUnit et de l'existence ou non de valeurs non nulles pour le taux de couverture, penchons-nous à présent sur l'effort de test et la couverture des tests. En effet, si le couplage (entrant et sortant à travers les métriques CBO, COC et FO) semble être une des caractéristiques déterminantes des classes pour lesquelles nous avons des valeurs du taux de couverture non nulles, les données que nous récolterons à la suite de cette étape pourraient nous permettre de mieux saisir le lien entre l'effort de test, quantifié à partir des métriques de test et du taux de couverture.

5.2. Hypothèse

Nous avons pu constater que des classes d'un système donné peuvent être couvertes par des tests bien qu'aucun test JUnit n'ait été développé pour celles-ci. L'explication que nous avons avancée se base sur les caractéristiques de ces classes, que nous avons regroupées au sein d'une catégorie, notée CAT-1 (cf. *chapitre 3* pour un rappel de l'acquisition des données, leur traitement et la catégorisation des classes orientées-objet). Parmi ces caractéristiques, il y a entre autres le couplage, à travers les valeurs de la métrique COC, mais également la taille (métrique LOC) et la complexité (WMPC). Concernant

toujours ces caractéristiques, Mockus et al [Mockus 09] ont émis l'observation selon laquelle un cas de test d'une classe particulière peut affecter la couverture des tests d'autres classes; en d'autres termes, le fait de tester une classe donnée, présentant certaines caractéristiques, pourrait avoir une portée atteignant d'autres classes qui, par le fait même, seraient elles aussi testées.

Nous pouvons par conséquent nous demander si la couverture des tests se trouve impactée par l'effort de test: en effet, si les tests JUnit développés pour des classes d'un système donné permettent d'atteindre d'autres classes, il serait légitime d'analyser l'effort de test requis à cet effet.

Ainsi, il serait intéressant de se pencher sur les corrélations entre l'effort de test quantifié par les métriques de test et le taux de couverture.

Toutefois, il est à préciser, que dans cette partie du mémoire, nous ne nous préoccupons que des classes pour lesquelles nous observons des valeurs non nulles du taux de couverture et pour lesquelles des tests unitaires JUnit ont été développés, en d'autres termes, les classes des catégories CAT-2.

Ainsi, notre expérimentation se résume à:

- Procéder à une analyse des statistiques descriptives des métriques de test pour chacun des systèmes JFREECHART, JODA, IO et POI;
- Procéder à une analyse des corrélations de Pearson et de Spearman entre les métriques de test et le taux de couverture pour les classes de la

catégorie CAT-2, pour chacun des quatre systèmes et discuter des résultats obtenus;

- Faire une analyse graphique des tendances qui se dégagent de la comparaison des valeurs des métriques de test et celles du taux de couverture.

5.3. Expérimentation

5.3.1 JFREECHART

Les données des métriques de test ont été obtenues grâce à l'outil Together de Borland et les résultats d'expérimentations nous proviennent de l'outil XLSTAT.

Dans une première étape de notre démarche expérimentale, nous allons procéder à une analyse des statistiques descriptives des métriques de test pour chaque système. Dans le cadre de cette analyse, nous nous pencherons uniquement sur les classes appartenant à la catégorie CAT-2, car il s'agit d'étudier les métriques de test, donc les métriques calculées sur les classes pour lesquelles des tests JUnit ont été développés.

Cette approche permettra d'identifier des caractéristiques particulières au niveau des données des métriques de test tel qu'on peut le noter dans les *tableaux 40, 41 et 42*:

Variable	Observation	Minimum	Maximum	Moyenne
Tass	223	0,000	143,000	18,090
Tdata	223	0,000	265,000	24,619
Tinvok	223	0,000	118,000	22,126
Tloc	223	0,000	635,000	91,628
WTcov	223	20,000	11550,000	765,381

Tableau 40 - Statistiques descriptives des données des métriques de test

En analysant les moyennes des métriques TAssert et TInvok, qui représentent respectivement le nombre d'invocations des méthodes d'une classe et le nombre de méthodes invoquées pour une classe, nous pouvons déjà noter qu'il y a plus de 22% de méthodes invoquées que d'invocations de méthodes (Tass = 18 ; Tinvok = 22) d'après le *tableau 40*.

À travers l'analyse des corrélations de Pearson et de Spearman entre les métriques de test et le taux de couverture pour chaque système, nous saurons s'il existe un lien entre l'effort de test et le taux de couverture.

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,625	0,702	0,846	0,670
Tdata	0,625	1	0,644	0,748	0,467
Tinvok	0,702	0,644	1	0,814	0,635
Tloc	0,846	0,748	0,814	1	0,625
WTcov	0,670	0,467	0,635	0,625	1

Tableau 41 - Corrélations de Spearman : métriques de test / taux de couverture

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,729	0,792	0,923	0,756
Tdata	0,729	1	0,485	0,736	0,414
Tinvok	0,792	0,485	1	0,837	0,804
Tloc	0,923	0,736	0,837	1	0,699
WTcov	0,756	0,414	0,804	0,699	1

Tableau 42 - Corrélations de Pearson : métriques de test / taux de couverture

Les premiers résultats obtenus à travers les *tableaux 41 et 42* permettent de constater que les corrélations entre les métriques de test et le taux de couverture sont toutes significatives (les cellules dont le contenu est présenté en gras contiennent des valeurs de corrélations significatives, tel que le calcule l'outil XLSTAT). Notons également que les corrélations les plus fortes avec le taux de couverture concernent les métriques Tass (dans le *tableau 41*, nous avons 0,670; dans le *tableau 42*, nous avons 0,756) et Tinvok (dans le *tableau 41*, nous avons 0,635; dans le *tableau 42*, nous avons 0,804). Tass donne le nombre d'invocations des méthodes assert JUnit se produisant dans le code d'une classe de test. Sachant que selon Bruntink, Tass permet de donner une idée sur la taille d'une suite de tests et la construction des cas de tests [Bruntink 04, Bruntink 06], nous pouvons comprendre la forte corrélation avec la couverture des tests. Tinvok, qui renseigne sur le nombre de méthodes invoquées par une classe de test, précise l'effort requis pour tester les méthodes d'une classe couplée à d'autres classes. Ainsi, ces 2 métriques sont toutes relatives aux appels de méthodes.

Par ailleurs, le développement de tests JUnit pour chaque appel de méthode nécessite un effort proportionnel au nombre de méthodes que possède une classe. Le système JFREECHART compte en moyenne 18 méthodes pour les classes de la catégorie CAT-2 (statistiques descriptives du *tableau 4*) pour une moyenne de 18 invocations de méthodes et 22 méthodes invoquées (*tableau 40*). Cela reviendrait à dire qu'en moyenne, un test JUnit a été écrit et développé pour chaque méthode, ce qui a permis d'invoquer environ 22% de méthodes en plus. Par ce constat, nous avons pu apporter un élément pour étayer l'hypothèse que l'effort de test pourrait avoir un impact sur la couverture des tests.

La plus faible corrélation avec le taux de couverture concerne la métrique Tdata qui informe sur le nombre d'objets créés lors des tests. Il semblerait donc que, bien que la corrélation avec le taux de couverture soit significative, Tdata influence le moins la couverture des tests.

L'analyse des droites des tendances du taux de couverture versus les valeurs des métriques de test confirmeront ou non nos constats suite à l'analyse des corrélations de Pearson et de Spearman.

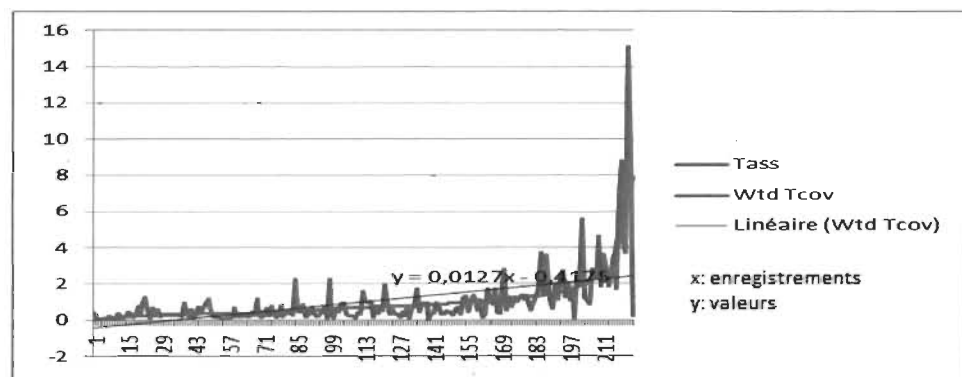


Figure 43 - Variation du taux de couverture vs la métrique de test Tass

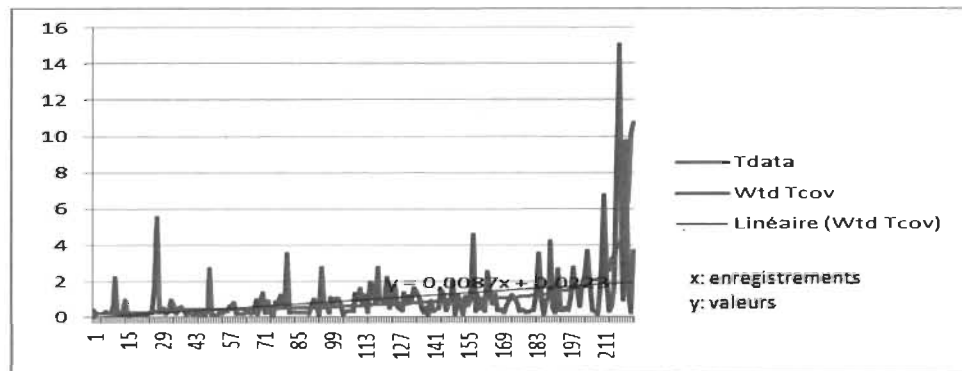


Figure 44 - Variation du taux de couverture vs la métrique de test Tdata

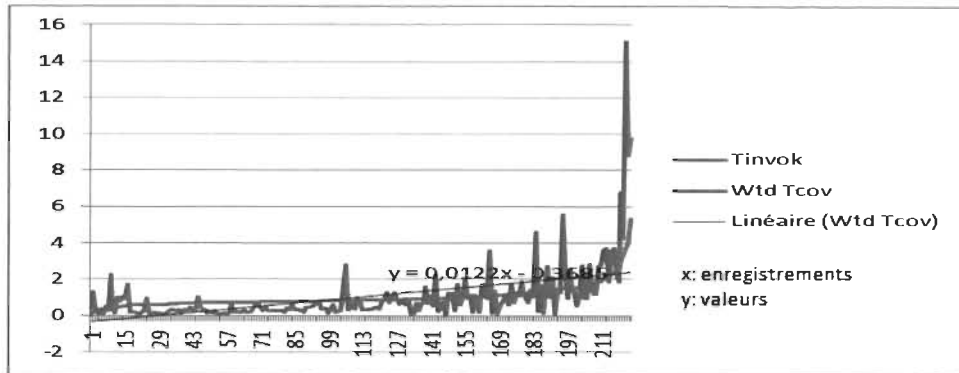


Figure 45 - Variation du taux de couverture vs la métrique de test Tinvok

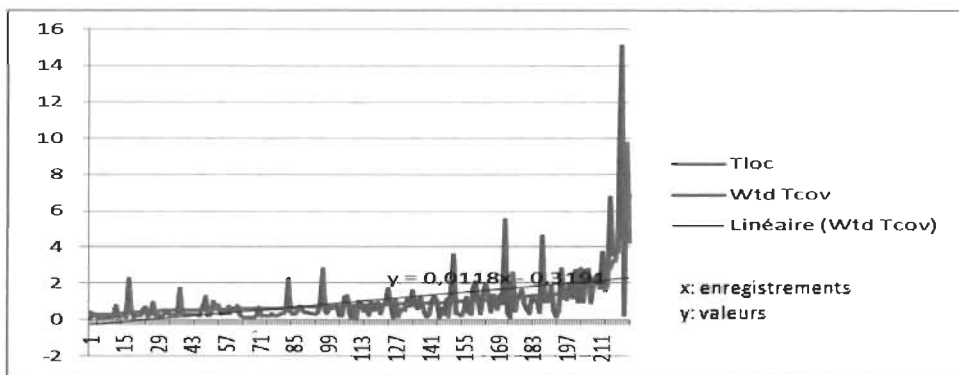


Figure 46 - Variation du taux de couverture vs la métrique de test Tloc

En se penchant sur les courbes de tendance des valeurs du taux de couverture en fonction des valeurs des métriques de test des *figures 43, 44, 45 et 46*, nous observons que les pentes des droites de tendance pour le cas des quatre métriques de test sont toujours positives, ce qui illustre bien la significativité des corrélations entre les métriques de test et le taux de couverture. En effet, cela signifie que si l'effort investi dans les tests augmente (effort, représenté par les valeurs des métriques de test) alors le taux de couverture augmente également. Une note particulière concernant la métrique Tass dans la *figure 43* (nombre d'invocations de méthodes) par rapport à laquelle nous avons la pente

de droite de tendance la plus haute (équation de la droite: $y = 0,0127x - 0,4175$).

Les corrélations entre les données des métriques de test et le taux de couverture sont significatives, en particulier pour les métriques Tass et Tinvok, métriques mettant en jeu des appels de méthodes. En comparant le nombre moyen de méthodes que compte le système JFREECHART pour les classes de la catégorie CAT-2 (seules classes pour lesquelles nous avons des valeurs pour les métriques de test, étant des classes pour lesquelles des tests JUnit ont été développés), le nombre d'invocations de méthodes lors des tests et le nombre de méthodes invoquées, nous réalisons le degré d'effort requis pour écrire un test pour chaque méthode d'une classe donnée. Cet effort qui aura permis d'atteindre environ 22% de méthodes en plus en termes de méthodes invoquées. De plus, une analyse des droites de tendance des valeurs du taux de couverture en fonction des métriques de test révèle que plus les valeurs de ces métriques de test augmentent, plus celles du taux de couverture augmentent également.

Ces résultats obtenus confortent l'hypothèse que l'effort de test peut impacter sur la couverture de tests et les résultats relevés sur les trois autres systèmes pourront nous aider à confirmer ou infirmer cette hypothèse.

5.3.2 JODA

De la même manière que nous avons procédé avec le système JFREECHART, dans une première étape de notre démarche expérimentale, nous allons procéder à une analyse des statistiques descriptives des métriques de test. Dans

le cadre de cette analyse, nous nous pencherons uniquement sur les classes appartenant à la catégorie CAT-2, car il s'agit d'étudier les métriques de test, donc les métriques calculées sur les classes pour lesquelles des tests JUnit ont été développés. Cette approche permettra d'identifier des caractéristiques particulières au niveau des données des métriques de test tel qu'on peut le noter dans les *tableaux 43, 44 et 45* suivants:

Variable	Observation	Minimum	Maximum	Moyenne
Tass	75	6,000	1156,000	225,573
Tdata	75	1,000	482,000	90,947
Tinvok	75	5,000	401,000	94,947
Tloc	75	27,000	2624,000	619,920
WTcov	75	66,000	9270,000	2229,040

Tableau 43 - Statistiques descriptives des données des métriques de tests

En analysant les statistiques descriptives du *tableau 43*, nous remarquons que le système JODA présente une taille non négligeable en termes de lignes de code au niveau des tests JUnit développés pour ses classes (moyenne: 620 lignes de codes).

À travers l'analyse des corrélations de Pearson et de Spearman entre les métriques de test et le taux de couverture pour chaque système, nous saurons s'il existe un lien entre l'effort de test et le taux de couverture.

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,803	0,917	0,950	0,666
Tdata	0,803	1	0,799	0,856	0,646
Tinvok	0,917	0,799	1	0,915	0,597
Tloc	0,950	0,856	0,915	1	0,688
WTcov	0,666	0,646	0,597	0,688	1

Tableau 44 - Corrélations de Spearman : métriques de test / taux de couverture

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,872	0,886	0,953	0,744
Tdata	0,872	1	0,857	0,930	0,717
Tinvok	0,886	0,857	1	0,923	0,735
Tloc	0,953	0,930	0,923	1	0,784
WTcov	0,744	0,717	0,735	0,784	1

Tableau 45 - Corrélations de Pearson : métriques de test / taux de couverture

L'effort requis pour l'écriture de ces tests est facilement concevable et lorsque nous nous penchons sur les données des *tableaux 44 et 45*, nous constatons que les corrélations entre les métriques de test et le taux de couverture sont toutes significatives et les valeurs maximales concernent les corrélations entre Tloc et WTcov (respectivement: 0,688 et 0,784 pour les corrélations de Spearman et Pearson). Tloc mesure le nombre de lignes de code d'une classe de test. Si une classe de test contient un nombre important de lignes de code, cela indique qu'un effort relativement important a été fourni pour écrire cette classe de test. Les tableaux de corrélations précédents (*tableaux 44 et 45*) nous indiquent que la taille des classes de test et la couverture des tests sont fortement corrélées.

L'analyse des droites des tendances du taux de couverture versus les valeurs des métriques de test confirmeront ou non nos constats suite à l'analyse des corrélations de Pearson et de Spearman.

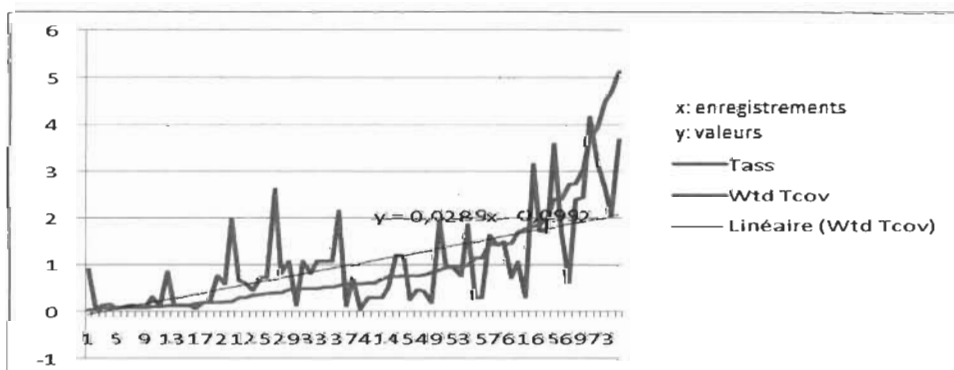


Figure 47 - Variation du taux de couverture vs la métrique de test Tass

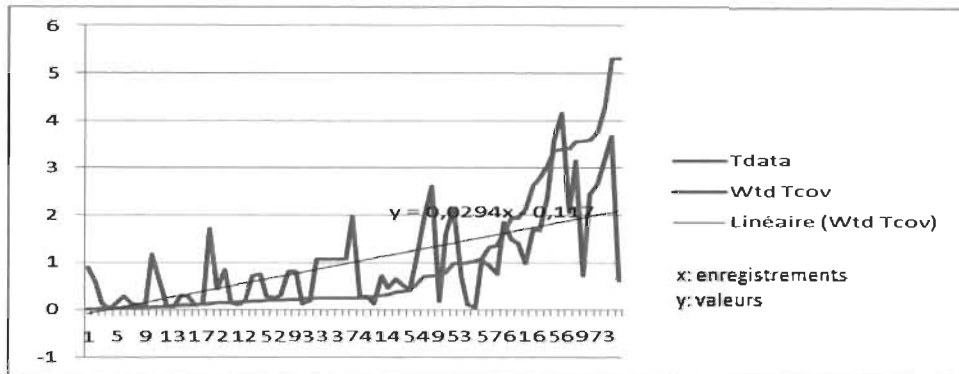


Figure 48 - Variation du taux de couverture vs la métrique de test Tdata

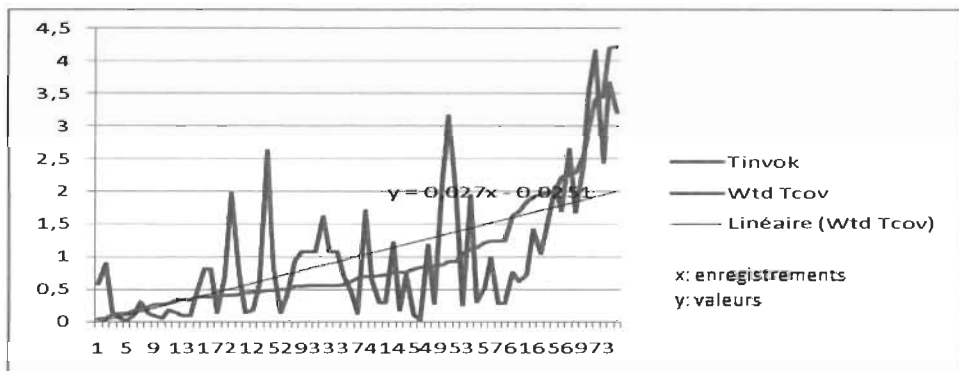


Figure 49 - Variation du taux de couverture vs la métrique de test Tinvok

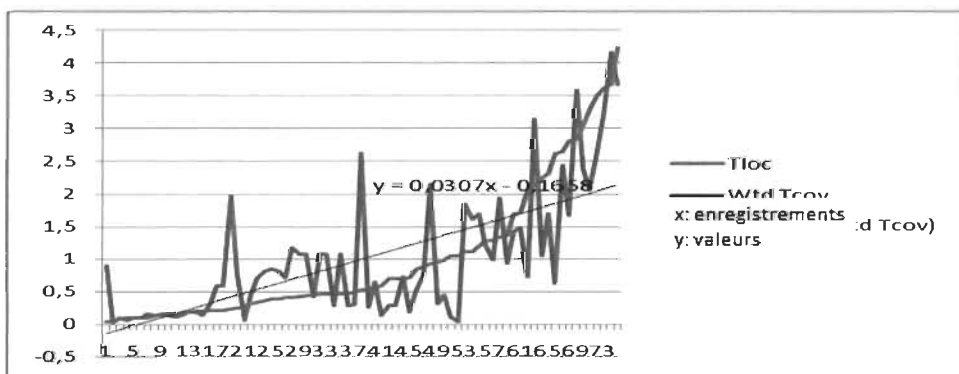


Figure 50 - Variation du taux de couverture vs la métrique de test Tloc

Les courbes de tendance des valeurs du taux de couverture en fonction des valeurs des métriques de test (*figures 47, 48, 49 et 50*), montrent que les pentes des droites de tendance pour le cas des quatre métriques de test sont toujours positives, ce qui illustre bien la significativité des corrélations entre les métriques de test et le taux de couverture. En effet, cela signifie que si l'effort investi dans les tests augmente (effort, représenté par les valeurs des métriques de tests) alors le taux de couverture augmente également. Une note particulière concernant la métrique Tloc (*figure 50*) par rapport à laquelle nous avons la pente de droite de tendance la plus haute (équation de la droite: $y = 0,0307x - 0,1658$). Cela donne une certaine idée de l'effort que représente d'écrire des tests comportant un nombre conséquent de lignes de code.

Ainsi, l'effort de test aurait effectivement un impact sur la couverture des tests.

5.3.3 IO

Comme avec les 2 précédents systèmes, la première étape de notre démarche expérimentale consiste en une analyse des statistiques descriptives des métriques de test. Cette approche permettra d'identifier des caractéristiques particulières au niveau des données des métriques de test tel qu'on peut le noter dans les *tableaux 46, 47 et 48*:

Variable	Observation	Minimum	Maximum	Moyenne
Tass	59	0,000	689,000	33,390
Tdata	59	0,000	219,000	15,068
Tinvok	59	0,000	145,000	17,915
Tloc	59	5,000	1600,000	165,153
WTcov	59	0,000	8184,000	836,373

Tableau 46 - Statistiques descriptives des données des métriques de tests

En analysant les statistiques descriptives du *tableau 46*, nous remarquons que le système IO présente une taille non négligeable en termes de lignes de code au niveau des tests JUnit développés pour ses classes (moyenne: 165 lignes de codes).

À travers l'analyse des corrélations de Pearson et de Spearman entre les métriques de test et le taux de couverture pour chaque système, nous saurons s'il existe un lien entre l'effort de test et le taux de couverture.

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,626	0,708	0,710	0,383
Tdata	0,626	1	0,764	0,796	0,445
Tinvok	0,708	0,764	1	0,852	0,653
Tloc	0,710	0,796	0,852	1	0,610
WTcov	0,383	0,445	0,653	0,610	1

Tableau 47 - Corrélations de Spearman : métriques de test / taux de couverture

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,360	0,406	0,607	0,484
Tdata	0,360	1	0,895	0,881	0,725
Tinvok	0,406	0,895	1	0,925	0,850
Tloc	0,607	0,881	0,925	1	0,798
WTcov	0,484	0,725	0,850	0,798	1

Tableau 48 - Corrélations de Pearson : métriques de test / taux de couverture

En analysant les tableaux de corrélations de Spearman et de Pearson (*tableaux 47 et 48*) entre les métriques de test et le taux de couverture, nous remarquons une corrélation de Pearson significative de 0,85 entre Tinvok et WTcov (*tableau 48*). Les corrélations les plus faibles avec le taux de couverture, bien que significatives, concernent la métrique de test Tassert pour laquelle nous avons une corrélation de Spearman de 0,383 (*tableau 47*) et une corrélation de Pearson de 0,484 (*tableau 48*). Ceci étant dit, comme nous l'avons constaté

avec les précédents systèmes, Tdata fait partie des métriques de test qui présentent les corrélations les moins fortes avec le taux de couverture.

L'analyse des droites des tendances du taux de couverture versus les valeurs des métriques de test confirmeront ou non nos constats suite à l'analyse des corrélations de Pearson et de Spearman.

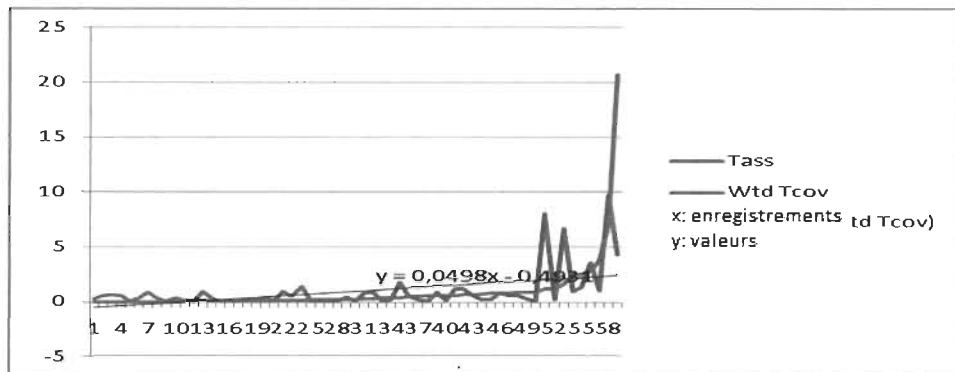


Figure 51 - Variation du taux de couverture vs la métrique de test Tass

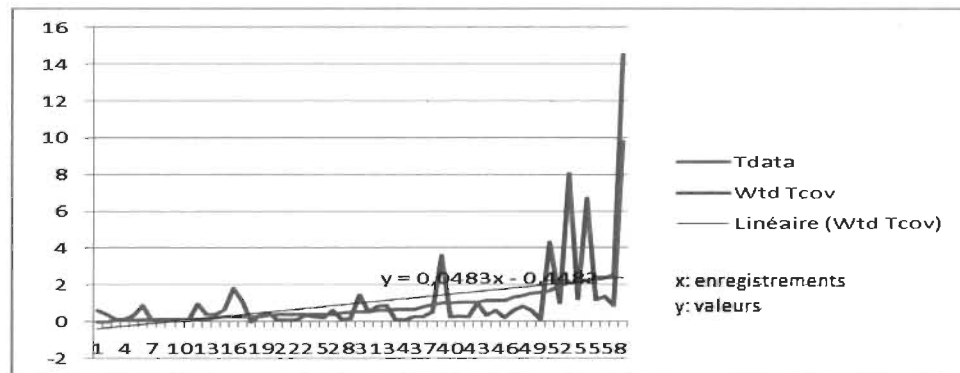


Figure 52 - Variation du taux de couverture vs la métrique de test Tdata

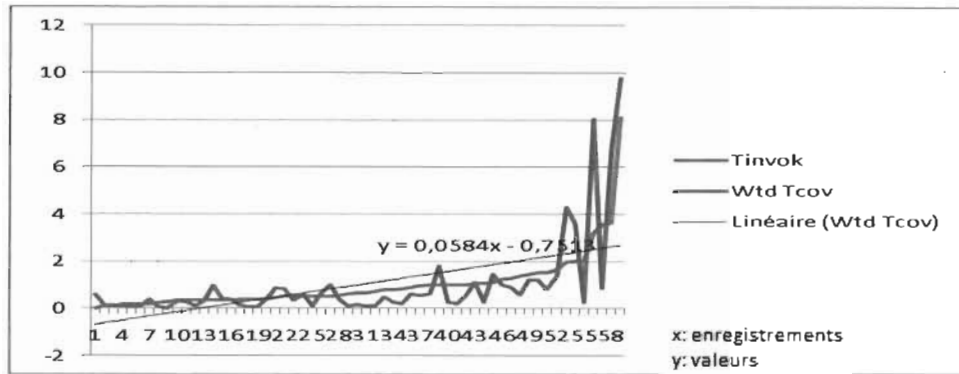


Figure 53 - Variation du taux de couverture vs la métrique de test Tinvok

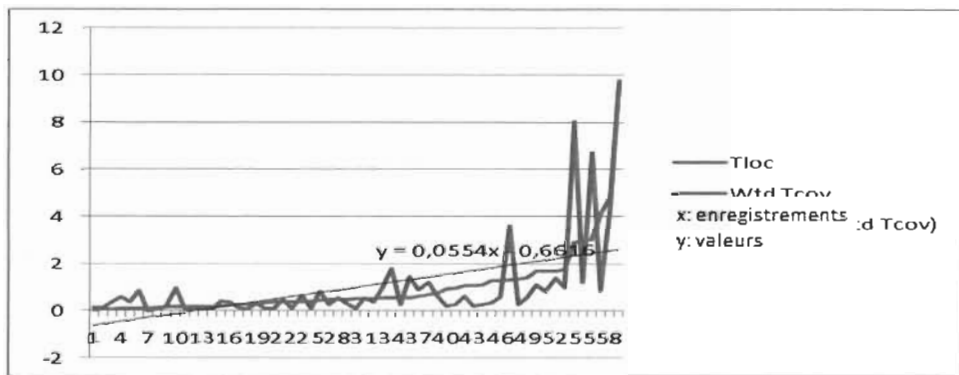


Figure 54 - Variation du taux de couverture vs la métrique de test Tloc

Les graphes de tendance des *figures 51, 52, 53 et 54* révèlent que les pentes des droites de tendance pour le cas des quatre métriques de test sont toujours positives, ce qui illustre bien la significativité des corrélations entre les métriques de test et le taux de couverture. En effet, cela signifie que si l'effort investi dans les tests augmente (effort, représenté par les valeurs des métriques de tests) alors le taux de couverture augmente également.

En particulier, Tinvok représente le nombre d'invocations de méthodes lors des tests et le graphe de la *figure 53* présente une pente positive (équation $y = 0,0584x - 0,7513$), illustrant bien que, plus il y a d'invocations de méthodes,

plus la couverture est grande. L'effort requis pour l'écriture des tests pour l'invocation de méthodes est quantifié par les valeurs de cette métrique au niveau des statistiques descriptives du système IO pour les classes de la catégorie CAT-2: Maximum = 145 et Moyenne = 17,915 (*tableau 46*) pour un nombre moyen de méthodes (métrique NOO) des classes orientées-objet de 10,12 (*tableau 22*).

Le système IO ne déroge pas à la règle: l'effort de test semble bien impacter positivement la couverture de test, ce qui permet d'établir un lien étroit entre l'effort de test et la couverture des tests.

5.3.4 POI

Comme avec les 3 précédents systèmes, la première étape de notre démarche expérimentale consiste en une analyse des statistiques descriptives des métriques de test. Cette approche permettra d'identifier des caractéristiques particulières au niveau des données des métriques de test tel qu'on peut le noter dans les *tableaux 49, 50 et 51*:

Variable	Observation	Minimum	Maximum	Moyenne
Tass	359	0,000	396,000	23,382
Tdata	359	0,000	185,000	7,819
Tinvok	359	0,000	138,000	21,206
Tloc	359	6,000	2379,000	105,343
WTcov	359	0,000	14288,000	1062,253

Tableau 49 - Statistiques descriptives des données des métriques de test

D'après le *tableau 49*, les moyennes des métriques TAssert et TInvok, qui représentent respectivement le nombre d'invocations des méthodes d'une classe et le nombre de méthodes invoquées pour une classe, montrent qu'il y a

presque autant de méthodes invoquées que d'invocations de méthodes (Tass = 23; Tinvok = 21).

À travers l'analyse des corrélations de Pearson et de Spearman entre les métriques de test et le taux de couverture pour chaque système, nous saurons s'il existe un lien entre l'effort de test et le taux de couverture.

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,444	0,673	0,729	0,431
Tdata	0,444	1	0,576	0,620	0,203
Tinvok	0,673	0,576	1	0,680	0,501
Tloc	0,729	0,620	0,680	1	0,311
WTcov	0,431	0,203	0,501	0,311	1

Tableau 50 - Corrélations de Spearman : métriques de test / taux de couverture

Variables	Tass	Tdata	Tinvok	Tloc	WTcov
Tass	1	0,599	0,597	0,789	0,602
Tdata	0,599	1	0,432	0,818	0,354
Tinvok	0,597	0,432	1	0,576	0,701
Tloc	0,789	0,818	0,576	1	0,447
WTcov	0,602	0,354	0,701	0,447	1

Tableau 51 - Corrélations de Pearson : métriques de test / taux de couverture

D'après les *tableaux 50 et 51*, il apparaît clairement que les corrélations entre les données des métriques de test et le taux de couverture sont toutes significatives autant pour la corrélation basée sur les rangs (Spearman) que pour la corrélation de Pearson et particulièrement pour les métriques Tass et Tinvok (Spearman: respectivement 0,431 et 0,501 dans le *tableau 50*; Pearson: respectivement 0,602 et 0,701 dans le *tableau 51*).

L'analyse des droites des tendances du taux de couverture versus les valeurs des métriques de test confirmeront ou non nos constats suite à l'analyse des corrélations de Pearson et de Spearman.

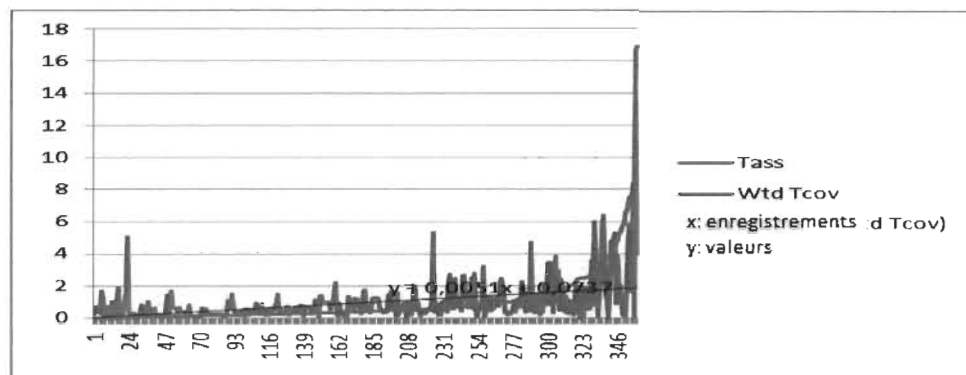


Figure 55 - Variation du taux de couverture vs la métrique de test Tass

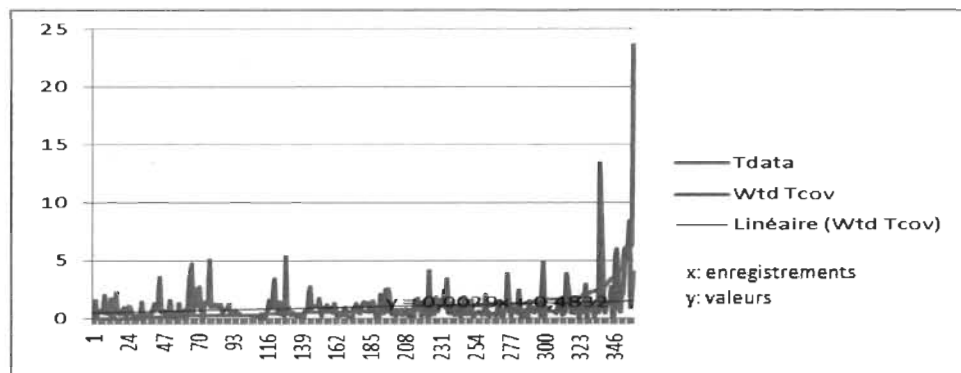


Figure 56 - Variation du taux de couverture vs la métrique de test Tdata

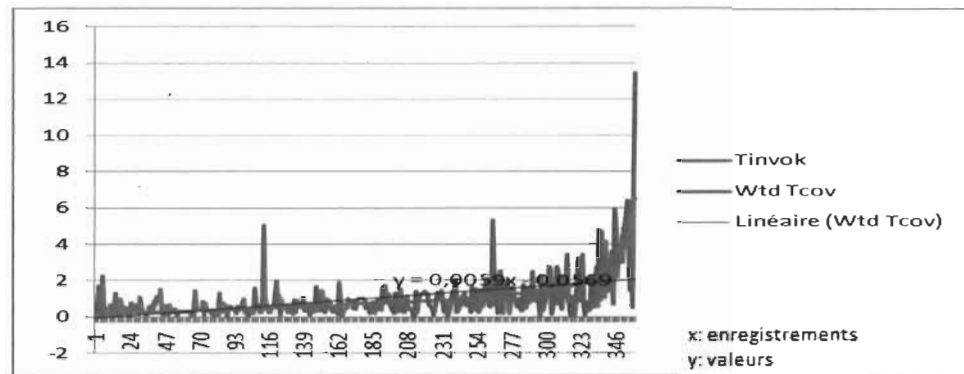


Figure 57 - Variation du taux de couverture vs la métrique de test Tinvok

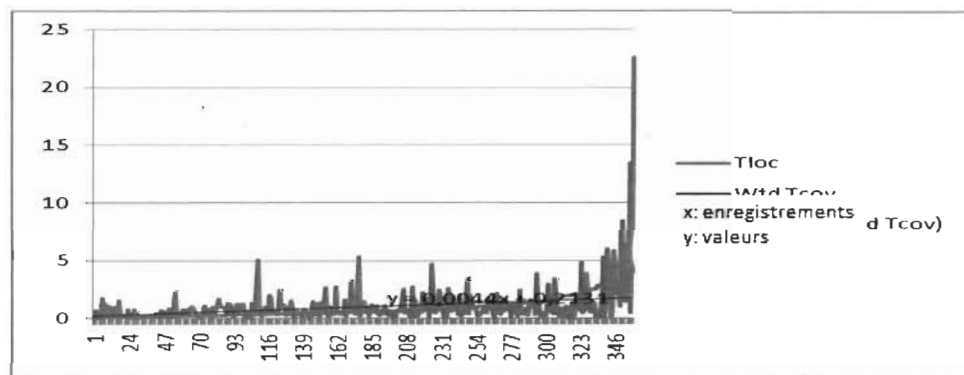


Figure 58 - Variation du taux de couverture vs la métrique de test Tloc

Les courbes des *figures 55, 56, 57 et 58* illustrent bien la significativité des corrélations entre les métriques de test et le taux de couverture. Les pentes des droites de tendances sont positives, ce qui signifie que si l'effort investi dans les tests augmente (effort, représenté par les valeurs des métriques de tests) alors le taux de couverture augmente également. Une note particulière concernant les métriques Tass et Tinvok (nombre d'invocations de méthodes) par rapport auxquelles nous avons les pentes de droites de tendances les plus hautes (équations de droites respectives: $y = 0,0051x + 0,0737$ et $y = 0,0059x - 0,0569$).

Les observations que nous avons faites tendent toutes vers le constat que l'effort de test impacte positivement la couverture des tests. Nous avons vu que plus les valeurs des métriques de test augmentent, plus il en est de même pour les valeurs du taux de couverture.

Sur les quatre systèmes que nous avons sélectionnés pour mener nos expérimentations, à savoir JFREECHART, JODA, IO et POI, les métriques Tinvok, Tass et Tloc sont celles pour lesquelles nous avons particulièrement noté que les corrélations avec le taux de couverture sont les plus fortes et les plus significatives: ces métriques permettent de quantifier l'effort relatif aux appels de méthodes (aussi bien le nombre d'invocations de méthodes que le nombre de méthodes invoquées) et le nombre de lignes de code. Les appels de méthodes nous ramènent, du point de vue des classes orientées-objet, aux métriques de couplage (CBO, FO et COC) et le nombre de lignes de code, aux métriques de taille et de complexité.

Ainsi, nous avons vu dans le *chapitre 4* que le couplage, la taille et la complexité (à travers les métriques COC, LOC et WMPC) sont des caractéristiques déterminantes des classes orientées-objet pour lesquelles nous avons des valeurs non nulles du taux de couverture, c'est-à-dire couvertes par les tests. Par ailleurs, à la suite de nos expérimentations au cours de ce chapitre, nous avons noté que l'effort de test relatif au couplage, la taille et la complexité impactent sur la couverture des tests.

Dans le chapitre suivant, nous allons tâcher de construire des modèles basés sur les métriques des classes orientées-objet pour prédire l'effort de test et par

la même occasion, prédire la couverture des test, et ceci dans l'optique d'une priorisation des tests.

CHAPITRE 6. VERS UNE PRIORISATION DES TESTS

6.1. Objectif

Nous avons précédemment vu qu'il existe un lien très significatif entre les métriques de test et le taux de couverture des tests: d'une manière générale, plus les valeurs des métriques de test augmentent, plus nous constatons un accroissement du taux de couverture.

Toutefois, la possibilité qu'un effort moindre et une couverture conséquente n'est pas à écarter. Nous orienterons le développement de cette étude dans ce sens afin de prioriser les tests de telle sorte que ceux-ci demanderaient un (relatif) faible effort d'écriture de test, pour une couverture de test maximale (élevée).

L'effort de test peut être en partie quantifié en se basant sur les valeurs des métriques de test. Nous avons également noté que les caractéristiques qui distinguent les classes pour lesquelles nous avons des valeurs du taux de couverture non nulles, que des tests JUnit aient été développés pour elles ou non, sont la taille (à travers la métrique LOC), le couplage (à travers la métrique COC) et la complexité (à travers la métrique WMPC).

Ainsi, au cours de ce chapitre, nous nous pencherons sur la maximisation de la couverture des tests en fonction des caractéristiques des métriques COC, LOC et WMPC tout en minimisant l'effort de test, et ce, dans le but de prioriser les tests.

6.2. Hypothèse

Nous chercherons durant cette étude à identifier les caractéristiques des classes qui vont permettre que l'effort de test soit minimal et la couverture des tests maximale. Ces caractéristiques seront basées sur les valeurs des métriques COC, LOC et WMPC et seront déterminées à partir d'une analyse des statistiques descriptives des classes dont les valeurs prédites de l'effort de test sont faibles et la couverture de test élevée. Le paragraphe suivant détaillera notre démarche expérimentale.

6.3. Démarche expérimentale

Nous savons que la taille (métrique LOC), le couplage (métrique COC) et la complexité (WMPC) sont des caractéristiques plus ou moins déterminantes pour la couverture des tests (*chapitre 4*).

Nous savons aussi que l'effort de test (à travers les métriques TLoc, TInvok, Tass) est fortement corrélé à la couverture des tests (*chapitre 5*).

Comme il n'existe pas de métrique particulièrement définie pour quantifier l'effort de test, nous opterons pour une approche qui va permettre de déterminer l'effort de test tel qu'il soit élevé ou faible. Ainsi, un effort de test élevé aura la valeur 1 et un effort de test faible 0. Nous préférons cet ensemble de valeurs (ensemble binaire) en vue de l'application d'une méthode statistique de prédiction basée sur des valeurs binaires: la régression logistique multivariée.

À l'aide de l'outil XLSTAT, il sera possible de calculer un pourcentage d'observations bien classées, pour un point de séparation (cutoff) donné. Pour une valeur de 0,5 du point de séparation, si la probabilité est inférieure à 0,5, l'observation est considérée comme étant affectée à la classe 0, sinon, elle est affectée à la classe 1.

Le calcul de l'effort de test devra utiliser les valeurs des métriques de test Tloc, Tinvok et Tass, métriques issues du chapitre précédent et qui ressortent de l'analyse des corrélations entre les métriques de test et le taux de couverture. Pour obtenir la valeur de l'effort de test pour une classe donnée, valeur qui sera comprise entre 0 et 1, soit une valeur binaire, nous procéderons comme suit:

- pour chacune des valeurs des métriques de test (Tloc, Tinvok et Tass) si une valeur est supérieure à la moyenne des valeurs d'une métrique, celle-ci sera remplacée par 1, sinon elle aura la valeur 0;
- après la binarisation des valeurs des métriques de test, le calcul de l'effort de test sera déterminé en fonction de la somme des valeurs binaires des métriques de test. Comme nous avons trois métriques, si la somme est égale à 3, alors l'effort sera considéré comme élevé sinon il sera considéré comme faible. Par conséquent pour chaque classe, pour lesquelles des tests JUnit ont été au niveau de chacun des quatre systèmes choisis dans le cadre de notre étude, nous serons en mesure de savoir si l'effort de test est élevé ou faible (soit 1 ou 0).

Une fois que l'effort de test a été binarisé, nous procéderons à la construction de modèles de prédiction en utilisant la régression logistique multivariée et en

prenant comme variable à expliquer l'effort de test et comme variables explicatives les métriques COC, LOC et WMPC. En effet, la régression logistique multivariée, telle que décrite au *chapitre 2*, est la méthode idéale pour construire un modèle de prédiction d'une variable binaire à partir d'un ensemble de valeurs de plusieurs variables. En prédisant efficacement l'effort de test, une priorisation des tests est envisageable en commençant par les tests nécessitant un moindre effort. Cette priorisation sera des plus optimales si les tests effectués permettent de couvrir ou atteindre d'autres classes. D'où la prochaine étape de notre démarche: prédire la couverture des tests.

Si nous désirons appliquer la régression logistique multivariée pour prédire la couverture des tests, il est nécessaire de binariser les valeurs du taux de couverture tel que la valeur 1 signifie, « une couverture élevée » et la valeur 0, « une couverture faible ». Ainsi, la variable à expliquer serait la couverture des tests. Par ailleurs, il nous faudra des variables explicatives, et suite à nos expérimentations décrites et commentées dans le *chapitre 4*, les variables explicatives désignées seraient les métriques orientées-objet COC, LOC et WMPC.

La binarisation du taux de couverture des tests se basera sur le calcul de la moyenne de la métrique WTcov; les valeurs se situant au-dessus de la moyenne correspondront à une couverture élevée, et celles en-dessous, une couverture (relativement) faible. Nous ne tiendrons compte ici que des classes pour lesquelles nous avons des valeurs non nulles du taux de couverture, à savoir les classes des catégories CAT-2 et CAT-1 (cf. chapitres 3 et 4).

La construction des modèles de prédiction aboutira donc sur 8 modèles dont 4 modèles de prédiction de l'effort de test et 4 modèles de prédiction de la couverture des tests.

Chacun des 2 modèles construits pour chacun des 4 systèmes utilisés au cours de cette étude, qui sont JFREECHART, JODA, IO et POI sera appliqué pour prédire l'effort de test d'une part et la couverture des tests d'autre part; cela résultera en un ensemble de données tel que nous aurons pour chaque classe de chaque système, 4 valeurs prédites de l'effort de test et 4 valeurs prédites de la couverture des tests.

Pour déterminer l'effort général de test prédit et la couverture générale des tests prédite, nous procéderons aux calculs suivants:

- pour l'effort de test, si la somme des valeurs binaires prédites par les 4 modèles est strictement supérieure à 2 alors l'effort général de test prédit sera 1 (soit élevé) sinon sera 0;
- de façon analogue, la couverture générale des tests prédite sera 1 (soit élevée) sinon 0 (faible).

Nous terminerons notre démarche expérimentale en commentant les résultats obtenus pour les classes dont l'effort de test prédit est faible alors que la couverture des tests prédite est élevée. Ces classes nous intéressent particulièrement dans le sens où leurs caractéristiques permettent d'atteindre l'objectif cité plus haut qui est de maximiser la couverture des tests en fonction

des caractéristiques des métriques COC, LOC et WMPC tout en minimisant l'effort de test, afin d'établir une priorisation des tests.

En effet, une classe dont l'effort de test prédit est faible et la couverture des tests prédite est élevée, est pour ainsi dire, idéale à tester, étant donné qu'en déployant un effort moindre pour développer les tests, nous pouvons atteindre une couverture importante. Ce cas de figure (moindre effort de test et couverture des tests élevée) devrait constituer un critère important, voire primordial, pour la priorisation des tests dans l'établissement des plans de tests.

Par ailleurs, si les classes répondant au critère primordial de priorisation des tests présentent des caractéristiques similaires (caractéristiques basées sur les valeurs des métriques COC, LOC et WMPC), alors nous serons en mesure d'effectuer la priorisation des tests à partir de ces caractéristiques.

6.4. Expérimentation

6.4.1 Régression logistique

La régression logistique multivariée telle que décrite au *chapitre 2* et dont l'application a été définie dans la démarche expérimentale, aboutit en un modèle de prédiction, caractérisé par une équation, dont les variables explicatives sont les métriques orientées-objet COC, LOC et WMPC. Une équation de modèle, dans le cadre d'une régression logistique, permet de prédire une valeur préalablement binarisée de la forme :

$$E = \frac{1}{1 + e^{-(a + bx + cy + dz)}} ,$$

Équation 2 - Équation du modèle de prédiction d'une régression logistique

où a, b, c et d sont des coefficients, x, y et z sont les valeurs respectives des métriques COC, LOC et WMPC pour une classe donnée et E est la valeur prédite (soit de l'effort de test, soit de la couverture des tests) avec E comprise entre 0 et 1. Si E est inférieur à 0,5, alors nous l'affecterons à 0 (soit effort de test ou couverture des tests faible), sinon 1 (soit effort de test ou couverture des tests élevée).

6.4.1.1. Stepwise

Stepwise est une procédure permettant de sélectionner la meilleure équation de régression logistique multivariée.

Le *Stepwise selection forward* est une procédure qui consiste à introduire une variable indépendante, pour ensuite, à chaque variable ajoutée, réévaluer si une variable devrait, selon le critère de sortie, être éliminée de l'équation. Le processus se termine lorsqu'il n'y a plus de variable respectant les critères d'entrée et de sortie. En résultat final, on obtient les variables indépendantes retenues comme étant les plus significatives par rapport au seuil fixé. Cette procédure présente l'avantage que seules les variables les plus pertinentes sont retenues tandis que celles ayant une faible contribution dans la prédiction de la variable à expliquer sont ignorées. De cette manière, on se retrouve avec un modèle composé de l'essentiel des variables retenues pour optimiser la précision et la qualité de la prédiction [Singh et al. 09].

Nous appliquerons cette méthode durant nos expérimentations pour comparer la sélection de métriques qu'elle propose, avec la sélection des métriques COC, LOC et WMPC dans la construction de nos modèles de prédiction. Pour ce faire, à partir d'un ensemble de variables explicatives constituées des métriques orientées-objet (CBO, COC, FO, NOO, DOIH, LOC, NOA, RFC, WMPC; nous ne retiendrons pas LCOM, qui présente plusieurs données manquantes), nous appliquerons la méthode Stepwise pour prédire l'effort de test et la couverture des tests et commenterons l'ensemble des métriques retenues pour les modèles de prédiction générés.

6.4.1.2. R^2 de McFadden

Il s'agit de vérifier si le modèle présente une vraisemblance (LLM) ou une log-vraisemblance (LL0) plus favorable. Plusieurs R^2 sont proposés dans la littérature et le R^2 de McFadden est l'un des plus simples à appréhender [Hosmer 00]. Lorsque la régression ne sert à rien, les variables explicatives n'expliquent rien, l'indicateur vaut 0. Lorsque la régression est parfaite, l'indicateur vaut 1. Menard suggère que le R^2 McFadden est le plus adapté à la régression logistique [Ménard 02].

6.4.1.3. Aire sous la courbe ROC

L'aire sous la courbe ROC (Receiver Operating Characteristic) [Singh et al. 09], indique la précision d'un modèle de prédiction et informe sur sa performance. On désigne par sensibilité (sensitivity) la proportion d'événements positifs bien classés. La spécificité (specificity) correspond à la proportion d'événements négatifs bien classés. Si l'on fait varier la probabilité seuil

(cutoff) à partir de laquelle on considère qu'un événement doit être considéré comme positif, la sensibilité et la spécificité varient. La courbe des points (1-spécificité, sensibilité) est appelée la courbe ROC. L'aire sous la courbe (ou Area Under the Curve - AUC) est un indice synthétique évalué pour les courbes ROC. L'AUC correspond à la probabilité pour qu'un événement positif ait une probabilité (donnée par le modèle) plus élevée qu'un événement négatif. Pour un modèle idéal, on a $AUC=1$, pour un modèle aléatoire, on a $AUC=0.5$. On considère habituellement que le modèle est bon dès lors que la valeur de l'AUC est supérieure ou égale à 0.7. Un modèle bien discriminant doit avoir une AUC entre 0.87 et 0.9. Un modèle ayant une AUC supérieure à 0.9 est excellent [Liu et al 05, Hall 90].

6.4.2 Interprétation des modèles

Avant d'entamer nos expérimentations, nous allons établir les équations des modèles de prédiction de l'effort de test d'une part, et celles de la couverture des tests d'autre part, pour chacun des systèmes JFREECHART, JODA, IO et POI. L'obtention des modèles s'est faite à l'aide de l'outil XLSTAT qui nous renseigne également sur les aires sous la courbe ROC, ainsi que les R^2 (notions définies précédemment), tels que présentés dans le *tableau 52*:

Test Effort	Test Coverage
$E_{TJF} = \frac{1}{1 + e^{-(3,2839 - 0,0161 \cdot COC - 0,0052 \cdot LOC + 0,0620 \cdot WMPC)}}$ AUC = 0,873 $R^2 = 0,303$ Stepwise : {FO, LOC}	$E_{CJF} = \frac{1}{1 + e^{-(3,2199 + 0,0338 \cdot COC - 0,0158 \cdot LOC + 0,1369 \cdot WMPC)}}$ AUC = 0,877 $R^2 = 0,334$ Stepwise : {NOO, DOIH, NOA}
$E_{TJD} = \frac{1}{1 + e^{-(5,8041 - 0,8882 \cdot COC - 0,0065 \cdot LOC + 0,1465 \cdot WMPC)}}$ AUC = 0,983 $R^2 = 0,705$ Stepwise : {DOIH, FO, LOC, NOA, WMPC}	$E_{CJD} = \frac{1}{1 + e^{-(3,7945 + 0,0429 \cdot COC + 0,0008 \cdot LOC + 0,0660 \cdot WMPC)}}$ AUC = 0,930 $R^2 = 0,437$ Stepwise : {FO, DOIH, NOO}
$E_{TIO} = \frac{1}{1 + e^{-(5,2270 + 0,6522 \cdot COC - 0,0262 \cdot LOC + 0,2151 \cdot WMPC)}}$ AUC = 0,958 $R^2 = 0,627$ Stepwise : {COC, NOO}	$E_{CIO} = \frac{1}{1 + e^{-(4,0231 + 0,5735 \cdot COC + 0,0404 \cdot LOC - 0,0299 \cdot WMPC)}}$ AUC = 0,892 $R^2 = 0,433$ Stepwise : {COC, NOO, RFC}
$E_{TPOI} = \frac{1}{1 + e^{-(3,0833 + 0,0032 \cdot COC - 0,0029 \cdot LOC + 0,0480 \cdot WMPC)}}$ AUC = 0,800 $R^2 = 0,234$ Stepwise : {DOIH, WMPC}	$E_{CPOI} = \frac{1}{1 + e^{-(3,0835 + 0,0221 \cdot COC + 0,0018 \cdot LOC + 0,0653 \cdot WMPC)}}$ AUC = 0,903 $R^2 = 0,316$ Stepwise : {COC, DOIH, FO, NOA, NOO, RFC, WMPC}

Tableau 52 - Équations de modèle obtenues pour les 4 systèmes

Le premier constat que nous pouvons d'ores et déjà faire à partir du *tableau 52* est que les 8 modèles construits à partir des métriques COC, LOC et WMPC, prises comme variables explicatives, présentent une aire sous la courbe ROC (AUC) supérieure à 0,80, ce qui dénote de la précision et la performance significatives de ceux-ci. Les AUC des modèles construits à partir des données de JODA sont les plus élevées (nous avons respectivement pour l'effort de test et la couverture des tests: 0,983 et 0,930).

Par rapport à la qualité de la régression, que nous pouvons analyser à partir des valeurs du R^2 de McFadden, il semble que les modèles de JODA et IO présentent les valeurs les plus élevées (entre 0,40 et 0,70), ce qui justifie les AUC élevées citées précédemment. Nous avons toutefois des valeurs du R^2 moindres pour les modèles des systèmes JFREECHART et POI (pour l'effort

de test nous avons respectivement 0,303 et 0,234 et pour la couverture des tests, 0,334 et 0,316).

Par conséquent, à ce stade de notre analyse des modèles, un certain « pattern » des modèles se dessine avec d'une part des modèles plus précis et plus performants que sont ceux qui sont relatifs aux systèmes JODA et IO, et d'autre part, des modèles moins précis et moins performants, ceux qui sont relatifs à JFREECHART et POI.

Cela pourrait s'expliquer par le fait que JODA et IO comptent moins de classes (respectivement 225 et 109) que JFREECHART et POI (respectivement 496 et 1539). Ceci étant dit, cela nous conforte dans notre démarche sachant que les systèmes autant que leurs modèles correspondants sont hétérogènes.

En ce qui concerne Stepwise, méthode que nous avons utilisée pour comparer la sélection de métriques qu'elle propose avec la sélection des métriques COC, LOC et WMPC pour la construction des modèles, nous constatons que les ensembles de métriques sélectionnées, pour construire le meilleur modèle de prédiction de l'effort de test et la couverture des tests, diffèrent énormément les uns des autres que ce soit au niveau de leur taille (par exemple, le modèle obtenu avec Stepwise pour prédire l'effort de test pour JFREECHART compte 2 métriques, à savoir FO et LOC, alors que celui que nous avons obtenu pour prédire la couverture des tests en compte 3, NOO, DOIH et NOA, d'après le *tableau 52*).

Néanmoins, nous pouvons noter la présence de la métrique NOO dans chaque modèle de prédiction de la couverture des tests. Cela s'explique par le fait que

les données de la métrique NOO ont été utilisées pour pondérer les valeurs du taux de couverture (mentionné au *chapitre 3*).

Ainsi, l'application de la méthode Stepwise ne nous permet pas de valider la sélection des métriques COC, LOC et WMPC pour la construction de nos modèles de prédiction. Cependant, la variété de sélections de métriques qu'elle propose pour construire le meilleur modèle pour la prédiction de l'effort de test et la couverture des tests, confirme le fait que les données des systèmes que nous avons sont réellement hétérogènes, d'où la nécessité de proposer un modèle basé sur les mêmes métriques, tel que nous le proposons dans cette étude et suite aux résultats que nous avons obtenus dans les *chapitres 4 et 5*.

6.4.3 JFREECHART

En appliquant la démarche expérimentale décrite plus haut sur les données du système JFREECHART, nous obtenons le tableau des statistiques descriptives (*tableau 54*) en ce qui concerne les classes pour lesquelles l'effort général de test prédit par les modèles est faible, alors que la couverture générale des tests prédite est élevée.

En guise de rappel, la détermination finale de l'effort général de test et de la couverture générale des tests prédits se fait comme suit, après avoir obtenu toutes les valeurs prédites de l'effort de test et du taux de couverture pour chaque classe de JFREECHART à partir des 8 modèles présentés dans le *tableau 52* (4 modèles de la colonne « Test Effort » et 4 modèles de la colonne « Test Coverage ») :

- si l'effort de test prédit par 2 ou plus des modèles sur les 4 que nous avons, est faible alors nous considérerons que l'effort général de test prédit, noté E_{TALL} , est faible sinon il est élevé;
- si la couverture des tests prédite par 3 ou plus des modèles sur les 4 que nous avons, est élevée alors nous considérerons que la couverture générale des tests prédite, notée E_{CALL} , est élevée sinon elle est faible.

L'échantillon des données de prédiction de JFREECHART présenté dans le *tableau 53* illustre notre procédure de détermination de l'effort général de test et de la couverture générale des tests prédits, données obtenues à partir des 8 équations de modèles présentées dans le *tableau 52*. Cette même procédure sera appliquée sur les systèmes JODA, IO et POI.

Resource	COC	LOC	WMPC	E_{TJF}	E_{TJD}	E_{TJO}	E_{TJCI}	E_{TALL}	E_{CJF}	E_{CJD}	E_{CJO}	E_{CJCI}	E_{CALL}	Priorize
BlockContainer	14	224	52	0	0	1	0	0	1	1	1	1	1	1
AbstractContentBlock	62	253	63	0	0	1	0	0	1	1	1	1	1	1
XYSeries	24	263	68	0	0	1	0	0	1	1	1	1	1	1
RectangleConstraint	46	211	49	0	0	1	0	0	1	1	1	1	1	1

DateAxis	3	1030	134	0	1	0	1	0	0	1	1	1	1	1
ChartFactory	33	1044	129	0	0	1	1	0	0	1	1	1	1	1
CategoryPlot	47	1591	388	1	0	1	1	1	1	1	1	1	1	0
Plot	48	424	106	1	0	1	1	1	1	1	1	1	1	0
XYPlot	52	2041	470	1	1	1	1	1	1	1	1	1	1	0
Minute	2	163	33	0	0	0	0	0	0	0	1	0	0	0
MarkerAxisBand	1	122	18	0	0	0	0	0	0	0	1	0	0	0
MeterPlot	2	627	121	1	1	1	1	1	1	1	1	1	1	0
CategoryAxis3D	1	91	15	0	0	0	0	0	0	0	0	0	0	0
BarRender3D	1	399	39	0	0	0	0	0	0	0	1	1	0	0
ImageTitle	0	178	28	0	0	0	0	0	0	0	1	0	0	0
MinMaxCategoryRender	0	215	38	0	0	0	0	0	0	0	1	0	0	0
LogarithmicAxis	0	526	106	1	1	1	1	1	1	1	1	1	1	0
XYDatasetTableModel	0	61	17	0	0	0	0	0	0	0	0	0	0	0
XYZDataset	4	4	2	0	0	0	0	0	0	0	0	0	0	0
YsSymbolic	1	5	3	0	0	0	0	0	0	0	0	0	0	0

Tableau 53 - Échantillon de données de prédiction de JFREECHART

Les 6 premières lignes du *tableau 53* représentent les classes qui seront priorisées car elles présentent un effort général de test prédit (E_{TALL}) faible et une couverture générale des tests prédite (E_{CALL}) élevée. Dans une deuxième phase de la priorisation, les classes pour lesquelles la E_{CALL} est élevée, pourraient être considérées.

Nous pouvons noter d'après les échantillons que pour les classes pour lesquelles E_{CALL} est élevée nous avons également des données des métriques COC, LOC et WMPC relativement élevées.

L'échantillon, qui est représentatif de la totalité des données de prédiction, confirme parfaitement un élément soulevé dans le *chapitre 5*, selon lequel un effort de test élevé va de pair avec une couverture élevée. Il n'existe aucun cas dans nos données tel qu'à un E_{TALL} élevé, il corresponde une E_{CALL} faible.

Néanmoins, le cas d'un E_{TALL} faible et d'une E_{CALL} élevée est envisageable, suivant notre hypothèse et notre critère de priorisation des tests.

Les statistiques descriptives des classes que nous prioriserons selon notre démarche expérimentale et celles du reste des classes de JFREECHART nous éclairerons plus dans le cadre de notre étude.

Variable	Observation	Minimum	Maximum	Moyenne
COC	54	0,000	88,000	9,685
LOC	54	37,000	1044,000	336,185
WMPC	54	9,000	134,000	63,889

Tableau 54 - Statistiques : effort de test faible - couverture des tests élevée

Variable	Observation	Minimum	Maximum	Moyenne
COC	442	0,000	101,000	4,100
LOC	442	2,000	2041,000	113,480
WMPC	442	0,000	470,000	23,593

Tableau 55 - Statistiques : reste des classes

Sur les 496 classes que compte JFREECHART, 54 (soit 10,88%) sont celles pour lesquelles l'effort général de test prédit est faible tandis que la couverture générale des tests prédite est élevée comme nous l'indiquent les données du *tableau 54*. Ces 54 classes se caractérisent par des valeurs moyennes élevées des métriques COC, LOC et WMPC.

Les valeurs moyennes des métriques des autres classes, représentant près de 90% de l'ensemble des classes de JFREECHART sont plus faibles malgré les maximums qui sont tous élevés, comme on peut le constater dans le *tableau 55*.

À ce stade de nos expérimentations, nous pouvons déjà remarquer qu'il existe des classes pour lesquelles il est concevable de fournir un effort général de test

relativement faible pour un résultat de couverture générale des tests élevée. Ces classes représenteraient près de 11% des classes que compte le système JFREECHART.

6.4.4 JODA

En appliquant la démarche expérimentale décrite plus haut sur les données du système JODA, nous obtenons le tableau des statistiques descriptives (*tableau 57*) en ce qui concerne les classes pour lesquelles l'effort général de test prédit (E_{TALL}) par les modèles est faible, alors que la couverture générale (E_{CALL}) des tests prédite est élevée.

L'échantillon des données de prédiction de JODA présenté dans le *tableau 56* suivant a été obtenu à partir des 8 équations des modèles présentées dans le *tableau 52*, de la même manière que nous avons procédé avec JFREECHART précédemment.

Resource	COC	LOC	WMPC	E_{TJF}	E_{TJD}	E_{TJD}	E_{TPOI}	E_{TALL}	E_{CJF}	E_{CJD}	E_{CJD}	E_{CPOI}	E_{CALL}	Priorize
Chronology	57	55	49	0	0	1	0	0	1	1	1	1	1	1
DateTimeUtils	49	158	38	0	0	1	0	0	1	1	1	1	1	1
FieldUtils	56	135	35	0	0	1	0	0	1	1	1	1	1	1
PeriodType	15	464	89	0	0	1	0	0	1	1	1	1	1	1
AssembledChronology	19	426	89	0	0	1	1	0	1	1	1	1	1	1

.DateTimeField	48	52	50	0	0	1	0	0	1	1	1	1	1	1
DateMidnight	1	359	89	1	1	1	1	1	1	1	1	1	1	0
DateTime	2	592	138	1	1	1	1	1	1	1	1	1	1	0
DateTimeZone	34	539	119	1	0	1	1	1	1	1	1	1	1	0
BaseDateTimeField	5	304	87	1	1	1	1	1	1	1	1	1	1	0
ISODateTimeFormat	18	917	176	1	0	1	1	1	1	1	1	1	1	0
LocalDate	4	621	149	1	1	1	1	1	1	1	1	1	1	0
AbstractInstant	7	132	38	0	0	1	0	0	1	0	1	0	0	0
AbstractInterval	1	145	37	0	0	0	0	0	0	0	1	0	0	0
AbstractPeriod	4	80	24	0	0	1	0	0	0	0	1	0	0	0
RuleSet	1	19	5	0	0	0	0	0	0	0	0	0	0	0
Zone	1	80	14	0	0	0	0	0	0	0	0	0	0	0
ZoneInfoProvider	0	147	29	0	0	0	0	0	0	0	1	0	0	0
Property	1	45	10	0	0	0	0	0	0	0	0	0	0	0
YearMonthDayProperty	1	52	12	0	0	0	0	0	0	0	0	0	0	0

Tableau 56 - Échantillon de données de prédiction de JODA

Les 6 premières lignes du *tableau 56* représentent les classes qui seront priorisées car elles présentent un effort général de test prédit (E_{TALL}) faible et une couverture générale des tests prédite (E_{CALL}) élevée. Dans une deuxième phase de la priorisation, les classes pour lesquelles la E_{CALL} est élevée, pourraient être considérées.

Nous pouvons noter d'après les échantillons que pour les classes pour lesquelles E_{CALL} est élevée nous avons également des données des métriques COC, LOC et WMPC relativement élevées, par rapport aux valeurs moyennes des statistiques descriptives du *tableau 58*.

Comme c'est le cas avec les données de JFREECHART, l'échantillon des données de JODA que nous avons est représentatif de la totalité des données de prédiction et confirme parfaitement un élément soulevé dans le *chapitre 5*, selon lequel un effort de test élevé va de pair avec une couverture élevée. Il

n'existe aucun cas dans nos données tel qu'à un E_{TALL} élevé, il correspondre une E_{CALL} faible.

Les statistiques descriptives des classes que nous prioriserons selon notre démarche expérimentale et celles du reste des classes de JODA nous éclairerons plus dans le cadre de notre étude.

Variable	Observation	Minimum	Maximum	Moyenne
COC	34	0,000	57,000	11,706
LOC	34	52,000	1760,000	366,118
WMPC	34	30,000	119,000	62,059

Tableau 57 - Statistiques : effort de test faible - couverture des tests élevée

Variable	Observation	Minimum	Maximum	Moyenne
COC	191	0,000	34,000	2,487
LOC	191	3,000	917,000	100,225
WMPC	191	1,000	176,000	22,806

Tableau 58 - Statistiques : reste des classes

Sur les 225 classes que compte JODA, 34 (soit 15,11%) se retrouvent dans la catégorie des classes pour lesquelles l'effort général de test prédit est faible et la couverture générale des tests élevée.

Ces 34 classes se caractérisent, tel que le démontrent les statistiques descriptives (*tableau 57*), par des valeurs moyennes élevées des métriques COC, LOC et WMPC et particulièrement pour la métrique COC (nous avons 11,706 versus 2,487).

Les valeurs moyennes des métriques des autres classes, représentant près de 85% de l'ensemble des classes de JODA sont plus faibles malgré un maximum

plus élevé pour la métrique WMPC (nous avons 176 dans le *tableau 58* contre 155 dans le *tableau 57*).

Nous pouvons appuyer le constat selon lequel il existe des classes pour lesquelles il est concevable de fournir un effort de test faible pour un résultat de couverture des tests élevé. Ces classes présentent des caractéristiques apparentes au niveau des valeurs des métriques COC, LOC et WMPC qui sont 3 fois plus élevées que la moyenne.

Comme observé au niveau des données du système JFREECHART précédemment, la priorisation des tests pourrait ainsi se faire en utilisant les données prédites à partir des 8 modèles que nous avons construits, ou bien, en sélectionnant les classes pour lesquelles nous avons des valeurs élevées des métriques COC, LOC et WMPC.

Ainsi, bien que les systèmes que nous étudions soient hétérogènes, tant au niveau de leur taille qu'au niveau des caractéristiques de leurs données, les résultats de l'application de notre démarche expérimentale sur les systèmes JFREECHART et JODA convergent et tendent vers les mêmes observations.

6.4.5 IO

Le tableau des statistiques descriptives concernant les classes pour lesquelles l'effort général de test prédit par les modèles est faible alors que la couverture générale des tests prédite est élevée pour le système IO (*tableau 60*), et le tableau des statistiques descriptives des classes restantes (*tableau 61*), suivent le même « pattern » que les observations faites précédemment, à savoir que les

valeurs moyennes des données du *tableau 60* sont nettement plus élevées que celles du *tableau 61* à l'exception des valeurs de la métrique COC.

L'échantillon des données de prédiction de IO présenté dans le *tableau 59* suivant a été obtenu à partir des 8 équations des modèles présentées dans le *tableau 52*, de la même manière que nous avons procédé avec JFREECHART et JODA précédemment.

Resource	COC	LOC	WMP	ET _{JF}	ET _{JD}	ET _{HO}	ET _{POI}	ET _{TALL}	EC _{JF}	EC _{JD}	EC _{HO}	EC _{POI}	EC _{TALL}	Priorize
FileAlterationObserver	1	185	55	0	1	1	0	0	1	1	1	1	1	1
XmlStreamReader	1	312	62	0	1	1	0	0	1	1	1	1	1	1
FileFilterUtils	2	194	60	0	0	1	0	0	1	1	1	1	1	1
FilenameUtils	4	490	145	1	1	1	1	1	1	1	1	1	1	0
FileUtils	8	968	250	1	1	1	1	1	1	1	1	1	1	0
IOUtils	8	512	136	1	1	1	1	1	1	1	1	1	1	0
EndianUtils	1	202	33	0	0	0	0	0	0	0	1	0	0	0
ClosedInputStream	0	7	1	0	0	0	0	0	0	0	0	0	0	0
SwappedDataInputStream	0	95	18	0	0	0	0	0	0	0	0	0	0	0
XmlStreamReaderException	1	36	7	0	0	0	0	0	0	0	0	0	0	0

Tableau 59 - Échantillon de données de prédiction de IO

Les 3 premières lignes du *tableau 59* représentent les classes qui seront priorisées car elles présentent un effort général de test prédit (ET_{TALL}) faible et une couverture générale des tests prédite (EC_{TALL}) élevée. Dans une deuxième phase de la priorisation, les classes pour lesquelles la EC_{TALL} est élevée, pourraient être considérées.

Nous pouvons noter d'après les échantillons, que pour les classes où la EC_{TALL} est élevée nous avons également des données des métriques COC, LOC et WMPC relativement élevées par rapport aux statistiques descriptives du *tableau 61*.

Comme c'est le cas avec les données de JFREECHART et JODA, l'échantillon des données de IO que nous avons, est représentatif de la totalité des données de prédiction, et confirme parfaitement un élément soulevé dans le *chapitre 5*, selon lequel un effort de test élevé va de pair avec une couverture élevée. Les cas de données telles qu'à un E_{TALL} élevé corresponde une E_{CALL} faible sont inexistants.

Les statistiques descriptives des classes que nous prioriserons selon notre démarche expérimentale et celles du reste des classes de IO, nous éclairerons plus dans le cadre de notre étude.

Variable	Observation	Minimum	Maximum	Moyenne
COC	6	0,000	2,000	0,667
LOC	6	185,000	1642,000	620,000
WMPC	6	55,000	62,000	59,000

Tableau 60 - Statistiques : effort de test faible - couverture des tests élevée

Variable	Observation	Minimum	Maximum	Moyenne
COC	103	0,000	11,000	0,806
LOC	103	4,000	1184,000	81,078
WMPC	103	1,000	250,000	16,534

Tableau 61 - Statistiques : reste des classes

Sur les 109 classes que compte IO, 6 (soit 5,82%) correspondent aux classes pour lesquelles l'effort général de test prédit est faible et la couverture générale des tests prédite est maximale.

Ces 6 classes se caractérisent, tel que le démontrent les statistiques descriptives (*tableaux 60 et 61*), par des valeurs moyennes élevées des métriques LOC et WMPC et particulièrement pour la métrique LOC (nous avons 620 versus 81,078).

Les valeurs moyenne de la métrique COC sont sensiblement les mêmes pour les 2 groupes de classes.

Les résultats observés à partir des données du système IO renforcent le constat selon lequel les classes pour lesquelles nous avons des valeurs prédites de l'effort de test comme étant faibles et la couverture des tests comme étant élevée se caractérisent par des valeurs élevées des métriques LOC et WMPC, bien que pour le cas de IO, les valeurs de la métrique COC soient moins déterminantes (voir *tableaux 60 et 61*).

6.4.6 POI

POI est le dernier système sur lequel nous appliquerons notre démarche expérimentale. Il se distingue des 3 précédents systèmes par le nombre très élevés de classes qu'il compte (1539). Les résultats que nous obtiendrons suite à l'analyse du tableau des statistiques descriptives des classes pour lesquelles l'effort de test prédit est faible alors que la couverture des tests est élevée (*tableau 63*), et le tableau des classes restantes, vont nous permettre de confirmer si effectivement des caractéristiques particulières distinguent ces 2 types de classes et quelles sont-elles?

L'échantillon des données de prédiction de POI présenté dans le *tableau 62* suivant a été obtenu à partir des 8 équations des modèles présentées dans le *tableau 52*.

Resource	COC	LOC	WMPC	ET _{IF}	ET _{UB}	ET _{IO}	ET _{POI}	ET _{ALL}	EC _{IF}	EC _{UB}	EC _{IO}	EC _{POI}	EC _{ALL}	Priorize
Sheet	43	257	65	0	0	1	0	0	1	1	1	1	1	1
TextRun	26	389	83	0	0	1	0	0	1	1	1	1	1	1
RecordContainer	42	147	36	0	0	1	0	0	1	1	1	1	1	1

RecordInputStream	180	253	57	0	0	1	0	0	1	1	1	1	1	1
POILogger	76	296	60	0	0	1	0	0	1	1	1	1	1	1
StringUtil	39	213	44	0	0	1	0	0	1	1	1	1	1	1
ExEmbed	67	210	49	0	0	1	0	0	1	1	1	1	1	1
XSSFSheet	29	1409	374	1	1	1	1	1	1	1	1	1	1	0
XSSFWorkbook	40	717	182	1	0	1	1	1	1	1	1	1	1	0
XWPFParagraph	13	368	121	1	0	1	1	1	1	1	1	1	1	0
BottomMarginRecord	4	641	142	1	1	1	1	1	1	1	1	1	1	0
CalcCountRecord	2	1631	295	1	1	1	1	1	1	1	1	1	1	0
CellRecord	6	666	148	1	1	1	1	1	1	1	1	1	1	0
EscherPropertyFactory	3	48	10	0	0	0	0	0	0	0	0	0	0	0
EscherSpgrRecord	3	86	15	0	0	0	0	0	0	0	1	0	0	0
EscherSplitMenuColors	2	90	15	0	0	0	0	0	0	0	1	0	0	0
UnknownEscherRecord	2	86	17	0	0	0	0	0	0	0	1	0	0	0
ExtractorFactory	1	116	31	0	0	0	0	0	0	0	1	0	0	0
HDGFLZW	2	161	15	0	0	0	0	0	0	0	1	0	0	0
Pointer	7	27	10	0	0	1	0	0	0	0	1	0	0	0

Tableau 62 - Échantillon de données de prédiction de POI

Les 6 premières lignes du *tableau 62* représentent les classes qui seront priorisées car elles présentent un effort général de test prédit (E_{TALL}) faible et une couverture générale des tests prédite (E_{CALL}) élevée.

Nous pouvons noter d'après les échantillons que pour les classes pour lesquelles E_{CALL} est élevée nous avons également des données des métriques COC, LOC et WMPC relativement élevées par rapport aux statistiques descriptives du *tableau 64*.

L'échantillon des données de POI que nous avons est représentatif de la totalité des données de prédiction et confirme parfaitement un élément soulevé dans le *chapitre 5*, selon lequel un effort de test élevé va de pair avec une couverture

élevée; il n'existe aucun cas dans nos données tel qu'à un E_{TALL} élevé, il corresponde une E_{CALL} faible.

Les statistiques descriptives des classes que nous prioriserons selon notre démarche expérimentale et celles du reste des classes de POI nous éclairerons plus dans le cadre de notre étude.

Variable	Observation	Minimum	Maximum	Moyenne
COC	91	0,000	200,000	26,956
LOC	91	9,000	1231,000	301,912
WMPC	91	7,000	148,000	58,495

Tableau 63 - Statistiques : effort de test faible - couverture des tests élevée

Variable	Observation	Minimum	Maximum	Moyenne
COC	1448	0,000	113,000	3,440
LOC	1448	2,000	1686,000	72,494
WMPC	1448	0,000	374,000	13,298

Tableau 64 - Statistiques : reste des classes

Sur les 1539 classes que compte POI, 91 (soit 5,91%) correspondent aux classes pour lesquelles les données prédites sont idéales, à savoir: un effort général de test faible et une couverture générale des tests élevée.

Notons que, pour le cas de POI qui est un système non négligeable de par le nombre des classes qui le constituent, s'il est envisageable de prioriser les tests afin d'en faire le moins possible ou plutôt de déployer le moins d'effort possible pour les écrire et pourtant avec une couverture conséquente, les gains (de temps et donc évidemment de coûts) seraient relativement importants.

Ces 91 classes se caractérisent, tel que le démontrent les statistiques descriptives, par des valeurs moyennes élevées des métriques COC, LOC et

WMPC et particulièrement pour la métrique COC (nous avons 26,956 versus 3,440, soit plus de 7 fois plus élevée) tels que le montrent clairement les *tableaux 63 et 64*.

Les 1448 classes restantes présentent des moyennes beaucoup plus faibles et parmi elles, un effort considérable devra être déployé pour le développement des tests JUnit. Une optimisation de l'effort de test et donc des coûts trouve tout son sens dans ce genre de cas de figure où une priorisation des tests s'avère nécessaire et devra avant tout considérer les classes pour lesquelles la couverture des tests est importantes alors que l'effort requis pour cela est moindre.

Ainsi, suite à nos expérimentations sur les données des 4 systèmes orientés-objet que sont JFREECHART, JODA, IO et POI, nous constatons que pour 5 à 10% des classes de chacun de ces systèmes, il est possible de trouver des valeurs faibles prédites de l'effort de test et des valeurs élevées prédites de la couverture des tests.

Ces classes se caractérisent par des valeurs moyennes relativement élevées des métriques COC, LOC et WMPC, comparées aux valeurs moyennes des métriques des autres classes.

Une priorisation des tests pourrait donc ainsi se faire, en vue de minimiser l'effort de test et maximiser la couverture des tests, en se basant sur les classes (5 à 10% des classes) pour lesquelles nous avons des valeurs relativement élevées des métriques COC, LOC et WMPC, par rapport aux autres classes.

La priorisation des tests pourrait également se faire en appliquant la méthode basée sur les modèles de prédiction de l'effort général de test et la couverture générale des tests que nous avons construits et utilisés durant nos expérimentations au cours de cette étude.

Nous avons vu dans l'interprétation des données du *tableau 52* que les 8 modèles, que nous avons construits à partir des données de 4 systèmes différant énormément les uns des autres, sont des modèles performants, fiables mais aussi hétérogènes, comme l'illustre bien la différence de leur qualité de régression à travers les valeurs de leurs R^2 de McFadden.

CHAPITRE 7. CONCLUSION

La qualité des applications informatiques est une préoccupation qui concerne autant les chercheurs œuvrant dans le domaine du génie logiciel que les firmes de développement de logiciels. La testabilité est un volet de la qualité logicielle qui traite des différents éléments caractérisant la capacité d'un système à être testé. L'effort de test s'inscrit dans le processus de test et croît de manière significative avec la complexité des systèmes. La couverture des tests est une résultante du processus de test et la garantie de la qualité d'un logiciel augmente avec l'ampleur de la couverture des tests.

Les métriques orientées-objet offrent des informations précises et pertinentes permettant de dresser le profil d'un système logiciel donné. Les métriques de test permettent d'une certaine manière de quantifier l'effort de test et la couverture des tests se mesure également à l'aide d'une métrique, le taux de couverture.

À travers ce mémoire, nous avons tâché de trouver une façon de prioriser les tests lors du processus de test afin d'optimiser l'effort tout en maximisant la couverture. Ainsi, les coûts rattachés aux tests s'en trouveraient réduits tandis que les objectifs en termes de couverture des tests atteints.

Notre démarche a consisté à analyser les caractéristiques des classes orientées-objet via les métriques orientées-objet (dont le taux de couverture) et les métriques de test en comparant 3 catégories de classes: les classes pour lesquelles nous avons des valeurs non nulles du taux de couverture et pour lesquelles des tests unitaires JUnit ont été développés; les classes pour

lesquelles nous avons des valeurs non nulles du taux de couverture bien qu'aucun test unitaire n'ait été développé pour celles-ci; et les classes pour lesquelles nous n'avons aucune valeur du taux de couverture et aucun test unitaire n'a été développé pour celles-ci. Cette démarche nous a permis de dégager 3 métriques, à savoir les métriques COC (métrique de couplage), LOC (taille) et W MPC (complexité) dont les corrélations avec le taux de couverture sont significatives; nous avons aussi pu établir un lien entre l'effort de test et le taux de couverture.

Sur la base de ces constats, nous avons construit des modèles de prédiction de l'effort de test et du taux de couverture en prenant comme variables explicatives les métriques citées précédemment et nous les avons appliquées sur les 4 systèmes logiciels que sont les systèmes JFREECHART, JODA, IO et POI. L'aboutissement de notre démarche a permis d'atteindre notre but: prioriser les tests de telle sorte que l'effort de test soit optimisé et la couverture des tests maximisée. Cette priorisation peut se faire de 2 façons: en se servant des modèles de prédiction que nous avons construits dont les équations sont fournies dans le chapitre 6 de ce mémoire; ou en identifiant les classes pour lesquelles les valeurs des métriques COC, LOC et W MPC sont significativement élevées.

Ainsi, l'application des démarches et des méthodes expérimentales qui ont fait leurs preuves dans la littérature et dans le cadre de nos travaux, a ouvert de nouvelles perspectives dans le domaine du génie logiciel tel que nous proposons les deux approches énoncées plus haut.

Des avenues que nous proposerions seraient l'exploration d'autres types de systèmes logiciels, les systèmes orientés-aspect, ou encore l'utilisation des méthodes tirées de l'intelligence artificielle comme les arbres de décision pour renforcer les prédictions des modèles que nous avons construits pour la priorisation des tests.

BIBLIOGRAPHIE

- [Aggarwal 06] Aggarwal, K. K., Singh, Y., Kaur, A., Malhotra, R., Empirical Study of Object-Oriented Metrics, 2006.
- [Badri 09] Badri, M., Badri, L., Touré, F., Empirical Analysis of Object-Oriented Design Metrics: Towards a New Metric Using Control Flow Paths and Probabilities, 2009.
- [Basili 96] Basili, V. R., Briand, L. C., Melo, W. L., "A Validation of Object-Oriented Design Metrics as Quality Indicators", IEEE Transactions on Software Engineering, vol. 22, no. 10, pp. 751-761, October 1996.
- [Bieman 95] Bieman, J. M., Kang, B. K., "Cohesion and Reuse in an Object-Oriented System", Proceedings of the ACM Symposium on Software Reusability, April 1995.
- [Bonett 00] G. Bonett, D., A. Wright, T., Sample size requirements for estimating Pearson, Kendall and Spearman correlations, Volume 65, Issue 1, pp 23-28 - March, 2000.
- [Briand 97] Briand, L. C., Daly, J., Wurs, J., "The dimensions of coupling in object-oriented systems", Proceedings of OOPSLA, 1997.
- [Briand 99] Briand, L. C., Wüst, J., Lounis, H., "Using Coupling Measurement for Impact Analysis in Object-Oriented

- Systems", Proc. of the IEEE International Conference on Software Maintenance (ICSM), pp. 475-482, Aug./Sept. 1999.
- [Bruntink 04] Bruntink, M., Deursen, A. V., "Predicting class testability using object-oriented metrics", Proceedings of the IEEE International Workshop on Source Code Analysis and Manipulation, pp. 136-145, 2004.
- [Bruntink 06] Bruntink, M., Van Deursen, A., "An empirical study into class testability", Journal of Systems and Software, vol. 79, no.9, pp. 1219- 1232, 2006.
- [Chae 00] Chae, H. S., Know, Y. R., Bae, D. H., "A cohesion measure for object- oriented classes", Software Practice and Experience, no. 30, 2000.
- [Chidamber 94] Chidamber, S. R., Kemerer, C. F., "A Metric Suite for Object Oriented Design", IEEE Transactions on Software Engineering, vol. 20, n°6, pp. 477-493, 1994.
- [De Tran-Cao 05] DE TRAN-CAO, Mesure de la complexité fonctionnelle des logiciels, Thèse (Ph.D.) - Université du Québec à Montréal, 2005.
- [Dreiseitl 02] Dreiseitl, S., Ohno-Machado, L., "Logistic regression and artificial neural network classification models: A

methodology review", Journal of Biomedical Informatics, 35, 352–359, 2002.

- [El Emam 01] El Emam, K., Benlarbi, S., Goel, N., Rai, S. N., "The confounding effect of class size on the validity of object-oriented metrics", IEEE Transactions on Software Engineering, 27(7): 630-650, 2001.

- [Good 09] Good, P., Robustness of Pearson correlation, 2009.

- [Hall 90] Hall, P., La Scala, B., "Methodology and algorithms of empirical likelihood", Int Stat Rev, 58: 109–127, 1990.

- [Henderson-Sellers 96] Henderson-Sellers, B., "Object-Oriented Metrics, Measures of Complexity", Prentice Hall, 1996.

- [Hosmer 00] Hosmer, D. W., Lemeshow, S., Applied Logistic Regression, 2000.

- [Io] <http://commons.apache.org/proper/commons-io/> (Consulté le 13/04/2014)

- [Jfreechart] <http://www.jfree.org/jfreechart/> (Consulté le 13/04/2014)

- [Joda] <http://www.joda.org/joda-time/> (Consulté le 13/04/2014)

- [Kaur 13] Kaur, S., Singh, S., Kaur, H., "A Quantitative Investigation Of Software Metrics Threshold Values At Acceptable Risk Level ", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 3, March, 2013.
- [Kout 11] Kout, A., Badri, M., Toure, F., An Empirical Analysis of a Testability Model for Object-Oriented Programs, 2011.
- [Larman 03] Larman, G., Applying UML and Design Patterns, An introduction to object-oriented analysis and design and the unified process, second edition, Prentice-Hall, 2003.
- [Larman 03] Larman G., Applying UML and Design Patterns, An introduction to object-oriented analysis and design and the unified process, second edition, Prentice-Hall, 2003.
- [Li 93] Li, W., Henry, S., "Object-Oriented metrics that predict Maintainability", Journal of Systems and Software, vol. 23, 1993.
- [Liu 05] Liu, H., Li, G., Cumberland, W. G., Wu, T., "Testing Statistical Significance of the Area under a Receiving Operating Characteristics Curve for Repeated Measures Design with Bootstrapping", Journal of Data Science, 257-278, March, 2005.
- [Ménard 02] Menard, S., Applied Logistic Regression Analysis, 2002.

- [Mockus 09] Mockus, A., Nagappan, N., Dinh-Trong, D. T., Test coverage and Post-Verification Defects: a Multiple Case Study, 2009.
- [Nasseri 10] Nasseri, E., Counsell, S., Tempero, E., "An empirical study of fan-in and fan-out in Java OSS", 8th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010), Proceedings of the 8th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010).
- [Poi] <http://poi.apache.org/> (Consulté le 13/04/2014)
- [Singh 09] Singh Y., et al., Empirical validation of object-oriented metrics for predicting fault proneness models, 2009.
- [Somerville 04] Sommerville, I., Software Engineering, 7th edition, Addison Wesley, 2004.
- [Together] <http://www.borland.com/products/together/> (Consulté le 13/04/2014)
- [Weka] <http://www.cs.waikato.ac.nz/ml/weka/> (Consulté le 13/04/2014)
- [Xlstat] <http://www.xlstat.com> (Consulté le 13/04/2014)

[Xlstat]

<http://www.xlstat.com> (Consulté le 13/04/2014)